ProSoft® TECHNOLOGY

Where Automation Connects.



ProTalk®

PTQ-104S Rev2

Quantum Platform

IEC 60870-5-104 Server for Quantum

February 22, 2013

USER MANUAL

Your Feedback Please

We always want you to feel that you made the right decision to use our products. If you have suggestions, comments, compliments or complaints about our products, documentation, or support, please write or call us.

ProSoft Technology
5201 Truxtun Ave., 3rd Floor
Bakersfield, CA 93309
+1 (661) 716-5100
+1 (661) 716-5101 (Fax)
www.prosoft-technology.com
support@prosoft-technology.com

Copyright © 2013 ProSoft Technology, Inc., All rights reserved.

PTQ-104S Rev2 User Manual

February 22, 2013

ProSoft Technology ®, ProLinx ®, inRAx ®, ProTalk®, and RadioLinx ® are Registered Trademarks of ProSoft Technology, Inc. All other brand or product names are or may be trademarks of, and are used to identify products and services of, their respective owners.

In an effort to conserve paper, ProSoft Technology no longer includes printed manuals with our product shipments. User Manuals, Datasheets, Sample Ladder Files, and Configuration Files are provided on the enclosed DVD, and are available at no charge from our web site: http://www.prosoft-technology.com

Content Disclaimer

This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither ProSoft Technology nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. Information in this document including illustrations, specifications and dimensions may contain technical inaccuracies or typographical errors. ProSoft Technology makes no warranty or representation as to its accuracy and assumes no liability for and reserves the right to correct such inaccuracies or errors at any time without notice. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of ProSoft Technology. All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components. When devices are used for applications with technical safety requirements, the relevant instructions must be followed. Failure to use ProSoft Technology software or approved software with our hardware products may result in injury, harm, or improper operating results. Failure to observe this information can result in injury or equipment damage.

© 2013 ProSoft Technology. All rights reserved.

Printed documentation is available for purchase. Contact ProSoft Technology for pricing and availability.

North America: +1.661.716.5100

Asia Pacific: +603.7724.2080

Europe, Middle East, Africa: +33 (0) 5.3436.87.20

Latin America: +1.281.298.9109

Information for ProTalk® Product Users

The statement "power, input and output (I/O) wiring must be in accordance with Class I, Division 2 wiring methods Article 501-10(b) of the National Electrical Code, NFPA 70 for installations in the U.S., or as specified in section 18-1J2 of the Canadian Electrical Code for installations within Canada and in accordance with the authority having jurisdiction".

The following or equivalent warnings shall be included:

A Warning - Explosion Hazard - Substitution of components may Impair Suitability for Class I, Division 2;
B Warning - Explosion Hazard - When in Hazardous Locations, Turn off Power before replacing Wiring Modules, and
C Warning - Explosion Hazard - Do not Disconnect Equipment unless Power has been switched Off or the Area is known to be Nonhazardous.
D Caution: The Cell used in this Device may Present a Fire or Chemical Burn Hazard if Mistreated. Do not Disassemble, Heat above 100°C (212°F) or Incinerate.

WARNING - EXPLOSION HAZARD - DO NOT DISCONNECT EQUIPMENT UNLESS POWER HAS BEEN SWITCHED OFF OR THE AREA IS KNOWN TO BE NON-HAZARDOUS.

AVERTISSEMENT - RISQUE D'EXPLOSION - AVANT DE DÉCONNECTER L'ÉQUIPEMENT, COUPER LE COURANT OU S'ASSURER QUE L'EMPLACEMENT EST DÉSIGNÉ NON DANGEREUX.

Class I, Division 2 GPs A, B, C, D

II 3 G

Ex nA IIC X

0° C <= Ta <= 60° C

II - Equipment intended for above ground use (not for use in mines).

3 - Category 3 equipment, investigated for normal operation only.

G - Equipment protected against explosive gasses.

Warnings

North America Warnings

A Warning - Explosion Hazard - Substitution of components may impair suitability for Class I, Division 2.
B Warning - Explosion Hazard - When in hazardous locations, turn off power before replacing or rewiring modules.
Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous.
C Suitable for use in Class I, Division 2 Groups A, B, C and D Hazardous Locations or Non-Hazardous Locations.

ATEX Warnings and Conditions of Safe Usage:

Power, Input, and Output (I/O) wiring must be in accordance with the authority having jurisdiction.

A Warning - Explosion Hazard - When in hazardous locations, turn off power before replacing or wiring modules.
B Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous.
C These products are intended to be mounted in an IP54 enclosure. The devices shall provide external means to prevent the rated voltage being exceeded by transient disturbances of more than 40%. This device must be used only with ATEX certified backplanes.
D DO NOT OPEN WHEN ENERGIZED.

Electrical Ratings

- Backplane Current Load: 1100 mA maximum @ 5 Vdc ± 5%
- Operating Temperature: 0°C to 60°C (32°F to 140°F)
- Storage Temperature: -40°C to 85°C (-40°F to 185°F)
- Shock: 30 g operational; 50 g non-operational; Vibration: 5 g from 10 to 150 Hz
- Relative Humidity: 5% to 95% (without condensation)
- All phase conductor sizes must be at least 1.3 mm(squared) and all earth ground conductors must be at least 4mm(squared).

Markings:

CSA/cUL
CSA CB Certified
ATEX

Important Notice:

	CAUTION: THE CELL USED IN THIS DEVICE MAY PRESENT A FIRE OR CHEMICAL BURN HAZARD IF MISTREATED. DO NOT DISASSEMBLE, HEAT ABOVE 100°C (212°F) OR INCINERATE. Maximum battery load = 200 µA. Maximum battery charge voltage = 3.4 VDC. Maximum battery charge current = 500 µA. Maximum battery discharge current = 30 µA.

Contents

Your Feedback Please 2
Information for ProTalk® Product Users 3
Warnings 3
Important Notice: 4

Guide to the PTQ-104S User Manual 9

1 Start Here 11

1.1 Hardware and Software Requirements 12
1.1.1 Package Contents 12
1.1.2 Quantum Hardware 12
1.1.3 PC and Software 13
1.2 Install ProSoft Configuration Builder Software 13
1.3 Setting Up the ProTalk Module 14
1.3.1 Install the ProTalk Module in the Quantum Rack 14
1.3.2 Connect the PC to the ProTalk Configuration/Debug Port 16
1.3.3 Ethernet Configuration 18

2 Configuring the Processor with Unity Pro 21

2.1 Create a New Project 22
2.2 Add the PTQ Module to the Project 24
2.3 Build the Project 25
2.4 Connect Your PC to the Processor 26
2.4.1 Connecting to the Processor with TCPIP 28
2.5 Download the Project to the Processor 28

3 Configuring the Processor with Concept 31

3.1 Information for Concept Version 2.6 Users 32
3.1.1 Installing MDC Configuration Files 32
3.2 Creating a New Project 33
3.3 Adding the PTQ Module to the Project 36
3.4 Setting the Time of Day 39
3.5 Saving the Project 40
3.6 Downloading the Project to the Processor 41

4 Configuring the Processor with ProWORX 45

5 Module Configuration 49

5.1 Using ProSoft Configuration Builder 50
5.1.1 Setting Up the Project 50
5.1.2 Set Module Parameters 53
5.2 [Backplane Configuration] 54
5.2.1 Read Register Start 55

5.2.2 Read Register Count ... 55
5.2.3 Write Register Start ... 55
5.2.4 Write Register Count ... 55
5.2.5 3X Register Start ... 55
5.2.6 4X Register Start ... 56
5.2.7 Module Name ... 56
5.2.8 Error Offset ... 56
5.2.9 Initialize Output Data ... 56
5.3 Backplane Data Exchange ... 56
5.3.1 Data Type Mapping and Addressing ... 58
5.3.2 Read Status (Block 9250) ... 64
5.3.3 Event Messages (9958) ... 65
5.3.4 Read Module's Time to Processor (9970) ... 67
5.3.5 Block 9971: Set Module Time ... 68
5.3.6 Block 9998 or 9999: Reboot Module ... 68
5.4 [SNTP CLIENT] ... 69
5.4.1 NTP Server IP Address ... 70
5.4.2 Time Zone ... 70
5.4.3 Use Daylight Savings Time ... 71
5.4.4 Database Register ... 71
5.5 [I104S] Section ... 71
5.6 [I104S] > [IEC-870-5-104] ... 72
5.6.1 Use IP List ... 72
5.6.2 Override StartDT ... 72
5.6.3 Clear Queue on Close ... 73
5.6.4 t0 Connection Timeout ... 73
5.6.5 t1 Timeout Set Value ... 73
5.6.6 t2 Timeout Set Value ... 73
5.6.7 t3 Timeout Set Value ... 73
5.6.8 k (maximum queue) ... 73
5.6.9 w (latest ack threshold) ... 74
5.6.10 Time DB Offset ... 74
5.6.11 Error Offset ... 74
5.6.12 Command Delay Timer ... 75
5.6.13 Freeze Start Type ... 75
5.6.14 Interval for Freeze ... 75
5.6.15 Common Address of ASDU ... 75
5.6.16 Cyclic Data Transmission ... 75
5.6.17 Select/Operate Timeout ... 75
5.6.18 Use ACTTERM with Setpoint ... 76
5.6.19 Use ACTTERM with step ... 76
5.6.20 Event Scan Delay ... 76
5.6.21 Spontaneous Clock Events ... 76
5.6.22 Set Priority Queues ... 76
5.6.23 Cyclic Set IV Time ... 77
5.6.24 IV Check Delay Time ... 78
5.6.25 IV Fail Count ... 79
5.6.26 Scan Events ... 79
5.6.27 Time Type ... 80
5.6.28 Use Recent ... 80
5.7 [I104S] > [IEC-870-5-104 Database] ... 82
5.7.1 Short Pulse Time ... 82
5.7.2 Long Pulse Time ... 82
5.7.3 Default Command Qualifier ... 83

5.7.4 Override Command Qualifier 83
5.7.5 Point Count 83
5.7.6 Sequence Flag 84
5.7.7 Parameter Offset 85
5.8 [I104S] > [IEC-870-5-104 IP ADDRESSES] 86
5.9 [I104S] > [M_SP_NA_1 104] 86
5.10 [I104S] > [M_DP_NA_1 104] 87
5.11 [I104S] > [M_ST_NA_1 104] 87
5.12 [I104S] > [M_BO_NA_1 104] 87
5.13 [I104S] > [M_ME_NA_1 104] 88
5.14 [I104S] > [M_ME_NB_1 104] 88
5.15 [I104S] > [M_ME_NC_1 104] 88
5.16 [I104S] > [M_IT_NA_1 104] 89
5.17 [I104S] > [C_SC_NA_1 104] 89
5.18 [I104S] > [C_DC_NA_1 104] 90
5.19 [I104S] > [C_RC_NA_1 104] 90
5.20 [I104S] > [C_BO_NA_1 104] 90
5.21 [I104S] > [C_SE_NA_1 104] 91
5.22 [I104S] > [C_SE_NB_1 104] 91
5.23 [I104S] > [C_SE_NC_1 104] 92
5.24 Group Codes 92
5.25 To Create Optional Comment Entries 93
5.26 To print a configuration file 94

6 Downloading the Project to the Module 95

6.1 Downloading via Serial Connection 96
6.2 Downloading the Project via Ethernet Connection 98

7 Hot Standby Support 101

7.1 Hot Standby Overview 102
7.1.1 Identical Configurations 102
7.1.2 Primary and Standby Controllers 102
7.1.3 System Components 103
7.1.4 Modicon Quantum Hot Standby with Unity and IEC Logic 104
7.1.5 Understanding System Scan Time in Modicon Quantum Hot Standby with Unity Systems 104
7.2 Setting Up the Modicon Quantum Hot Standby with Unity System 106
7.2.1 Overview 106
7.2.2 Mapping the Backplane Extensions 107
7.2.3 PTQ-104S Hot Standby Considerations 107

8 Diagnostics and Troubleshooting 111

8.1 The Configuration/Debug Menu 112
8.1.1 Required Hardware 112
8.1.2 Using the Diagnostic Window in ProSoft Configuration Builder 112
8.1.3 Navigation 115
8.1.4 Main Menu 116
8.2 LED Indicators 121
8.2.1 Ethernet LED Indicators 122
8.2.2 Error Status Table 122

9 Reference 123

9.1 Product Specifications ... 124
9.1.1 Standards ... 124
9.1.2 General Specifications ... 124
9.1.3 Hardware Specifications ... 125
9.1.4 Functional Specifications ... 125
9.2 PTQ-104S Protocol Implementation ... 127
9.2.1 Module Address ... 127
9.2.2 Monitor Direction and Control Direction: Point Definition ... 130
9.2.3 Using Monitor Points ... 132
9.2.4 Using Control (Command) Points ... 143
9.2.5 Data Communication ... 150
9.2.6 Events ... 156
9.2.7 Sequence Flag ... 169
9.3 Cable Connections ... 170
9.3.1 Ethernet Connection ... 170
9.3.2 RS-232 Configuration/Debug Port ... 170
9.4 Error Status Table ... 171
9.5 Group Codes ... 174
9.6 IEC 60870-5-104 Server Interoperability Statement ... 175
9.6.1 System or device ... 175
9.6.2 Application Layer ... 175
9.6.3 Selection of standard ASDUs ... 176
9.6.4 Type identifier and cause of transmission assignments ... 179
9.6.5 Basic Application Functions ... 181
9.7 PTQ-104S Database Design Forms ... 186
9.7.1 M_SP_NA_1, M_DP_NA_1, M_ST_NA_1 and M_IT_NA_1 Form ... 186
9.7.2 M_ME_NA_1 and M_ME_NB_1 Form ... 187
9.7.3 Form for All C_ (Command) Data Types, Except C_RC_NA_1 ... 188
9.7.4 C_RC_NA_1 Form ... 189
9.8 Frequently Asked Questions ... 190
9.8.1 How fast do the "Backplane Data Exchange" commands run? ... 190
9.8.2 What is the maximum number of words I can transfer with a "Backplane Data Exchange" command? ... 190
9.8.3 How do I configure the module? ... 190
9.8.4 What kind of data transfer rates can I expect between the PLC and the module? 190
9.8.5 Is a .MDC available for configuration of the Module? ... 190
9.8.6 Does the module work in a remote rack? ... 190
9.8.7 Can I use the module in a hot backup system? ... 190

10 Support, Service & Warranty 191

10.1 Contacting Technical Support ... 191
10.2 Warranty Information ... 192

Index 193

Guide to the PTQ-104S User Manual

Introduction (Must Do)	→	Start Here (page 11)	This section introduces the customer to the module. Included are: package contents, system requirements, hardware installation, and basic configuration.
Diagnostic and Troubleshooting	→	Diagnostics and Troubleshooting (page 111)	This section describes Diagnostic and Troubleshooting procedures.
Reference Product Specifications Functional Specifications	→	Reference (page 123) Product Specifications (page 124) Functional Specifications (page 125)	These sections contain general references associated with this product, Specifications, and the Functional Overview.
Support, Service, and Warranty Index	→	Support, Service and Warranty (page 191) Index (page 193)	This section contains Support, Service and Warranty information. Index of chapters.

1 Start Here

In This Chapter

- Hardware and Software Requirements 12
- Install ProSoft Configuration Builder Software 13
- Setting Up the ProTalk Module 14

Note: The PTQ-104S Rev2 manual is intended for v2.07.000 and newer firmware. For v1.25.000 and older firmware, please use the PTQ-104S Rev1 manual.

Verifying the firmware of the PTQ-104S module can be found on page 119.

This guide is intended to guide you through the ProTalk module setup process, from removing the module from the box to exchanging data with the processor. In doing this, you will learn how to:

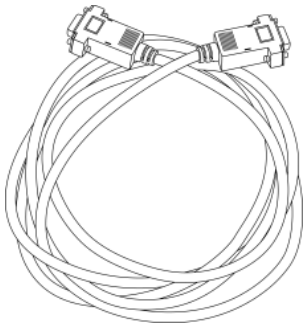
- Set up the processor environment for the PTQ module
- View how the PTQ module exchanges data with the processor
- Edit and download configuration files from your PC to the PTQ module
- Monitor the operation of the PTQ module

1.1 Hardware and Software Requirements

1.1.1 Package Contents



ProTalk Module



Null Modem Serial Cable



ProSoft Solutions DVD

1.1.2 Quantum Hardware

This guide assumes that you are familiar with the installation and setup of the Quantum hardware. The following should be installed, configured, and powered up before proceeding:

- Quantum Processor
- Quantum rack
- Quantum power supply
- Quantum Modbus Plus Network Option Module (NOM) (optional)
- Quantum to PC programming hardware
- NOM Ethernet or Serial connection to PC

1.1.3 PC and Software

- Windows-based PC with at least one COM port
- Quantum programming software installed on machine

  or
- Concept™ PLC Programming Software version 2.6

  or

  ProWORX PLC Programming Software

  or

  Unity™ Pro PLC Programming Software

Note: ProTalk modules are compatible with common Quantum programming applications, including Concept and Unity Pro. For all other programming applications, please contact technical support.

1.2 Install ProSoft Configuration Builder Software

This manual is meant for the use of the PTQ-104S Rev2 module. PCB version 4.1.0 and newer must be used for Rev2. All versions of PCB can be found at *http://www.prosoft-technology.com/pcb*

ProSoft Configuration Builder (PCB) software is used to configure the module.

To install ProSoft Configuration Builder from the ProSoft Web Site

1. Open your web browser and navigate to *http://www.prosoft-technology.com/pcb*
2. Click the **DOWNLOAD HERE** link to download the appropriate version of ProSoft Configuration Builder.
3. Choose **SAVE** or **SAVE FILE** when prompted.
4. Save the file to your *Windows Desktop*, so that you can find it easily when you have finished downloading.
5. When the download is complete, locate and open the file, and then follow the instructions on your screen to install the program.

If you do not have access to the Internet, you can install *ProSoft Configuration Builder* from the ProSoft Solutions DVD, included in the package with your module.

To install ProSoft Configuration Builder from the Product DVD

1. Insert the ProSoft Solutions Product DVD into the DVD drive of your PC. Wait for the startup screen to appear.
2. On the startup screen, click **PRODUCT DOCUMENTATION**. This action opens a Windows Explorer file tree window.
3. Click to open the **UTILITIES** folder. This folder contains all of the applications and files you will need to set up and configure your module.

4 Double-click the **SETUP CONFIGURATION TOOL** folder, double-click the **PCB_*.EXE** file and follow the instructions on your screen to install the software on your PC. The information represented by the "*****" character in the file name is the PCB version number and, therefore, subject to change as new versions of PCB are released.

Note: Many of the configuration and maintenance procedures use files and other utilities on the DVD. You may wish to copy the files from the Utilities folder on the DVD to a convenient location on your hard drive.

1.3 Setting Up the ProTalk Module

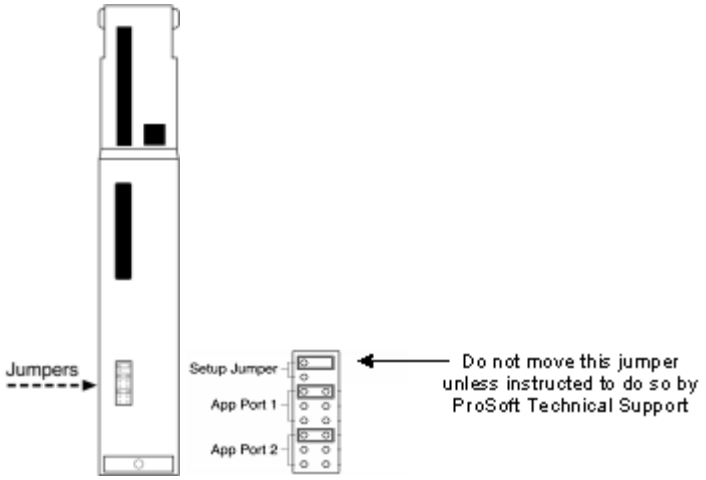
After you complete the following procedures, the ProTalk module will actively be transferring data bi-directionally with the processor.

1.3.1 Install the ProTalk Module in the Quantum Rack

Verify Jumper Settings

ProTalk modules are configured for RS-232 serial communications by default. To use RS-422 or RS-485, you must change the jumpers.

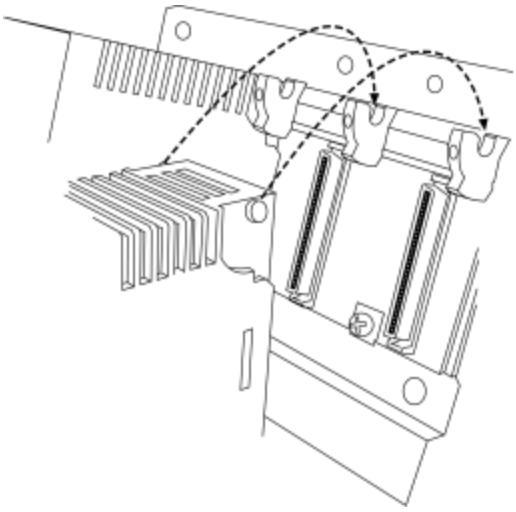
The jumpers are located on the back of the module as shown in the following illustration:



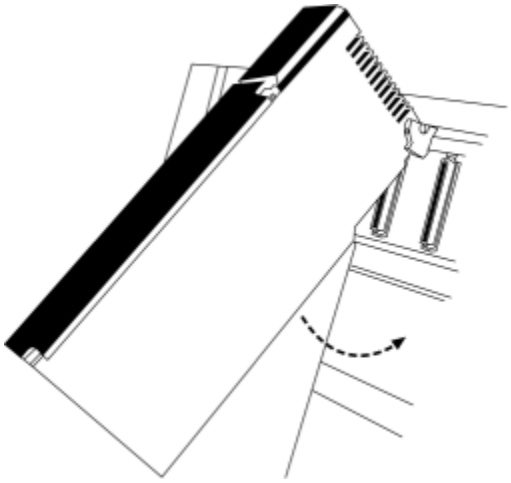
Install the ProTalk Module in the Quantum Rack

1 Place the Module in the Quantum Rack. The ProTalk module must be placed in the same rack as the processor.

2 Tilt the module at a 45° angle and align the pegs at the top of the module with slots on the backplane.



3 Push the module into place until it seats firmly in the backplane.

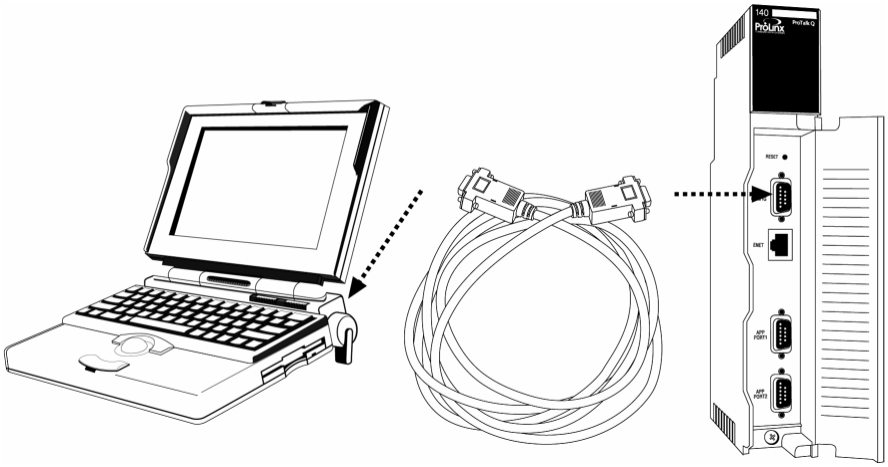


Caution: The PTQ module is hot-swappable, meaning that you can install and remove it while the rack is powered up. You should not assume that this is the case for all types of modules unless the user manual for the product explicitly states that the module is hot-swappable. Failure to observe this precaution could result in damage to the module and any equipment connected to it.

1.3.2 Connect the PC to the ProTalk Configuration/Debug Port

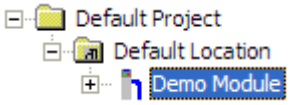
Make sure you have exited the Quantum programming software before performing these steps. This action will avoid serial port conflict.

Using the supplied Null Modem cable, connect your PC to the Configuration/Debug port on the ProTalk module as shown

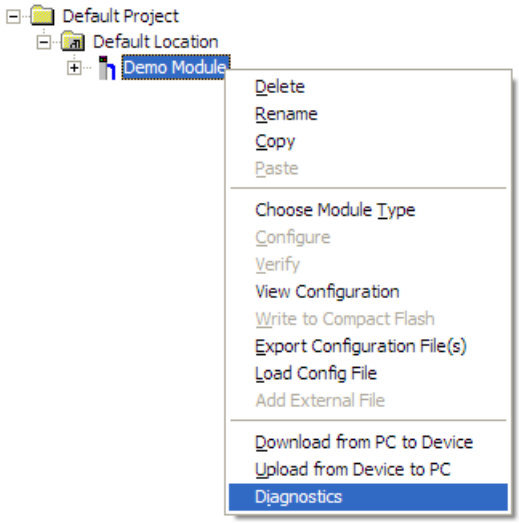


To connect to the module's Configuration/Debug serial port

1 Start PCB, and then select the module to test. Click the right mouse button to open a shortcut menu.

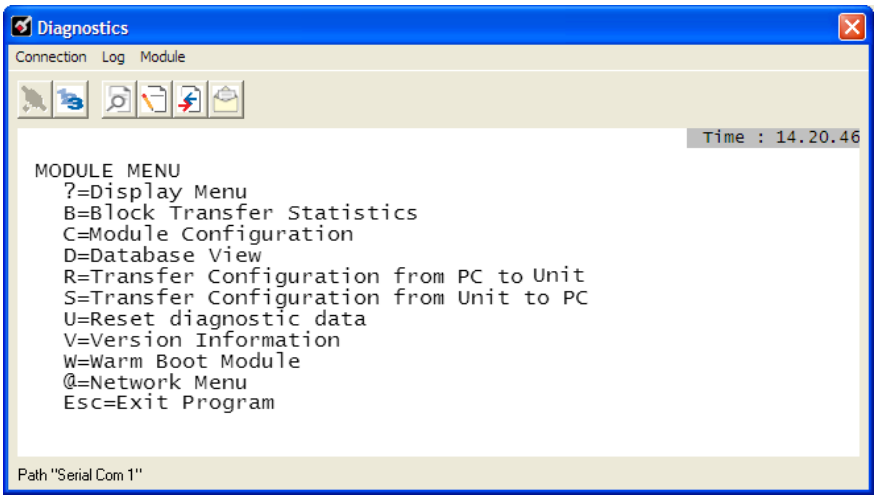


2 On the shortcut menu, choose **DIAGNOSTICS.**



This action opens the **DIAGNOSTICS** dialog box.

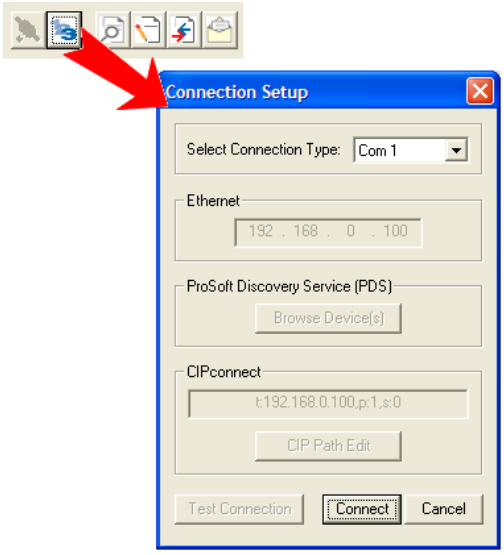
3 Press **[?]** to open the Main Menu.



Important: The illustrations of configuration/debug menus in this section are intended as a general guide, and may not exactly match the configuration/debug menus in your own module.

If there is no response from the module, follow these steps:

1 Click to configure the connection. On the Connection Setup dialog box, select a valid com port or other connection type supported by the module.



2 Verify that the null modem cable is connected properly between your computer's serial port and the module. A regular serial cable will not work.

3 On computers with more than one serial port, verify that your communication program is connected to the same port that is connected to the module.

4 If you are still not able to establish a connection, contact ProSoft Technology for assistance.

1.3.3 Ethernet Configuration

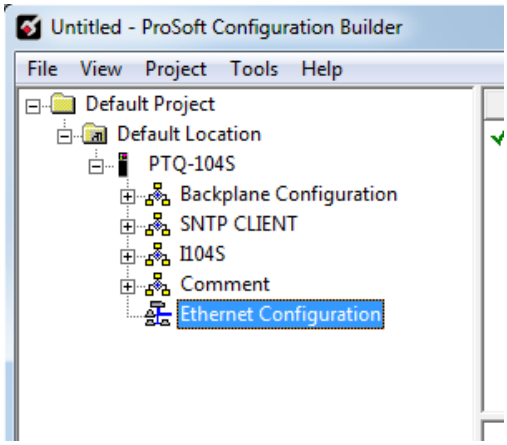
Use this procedure to configure the Ethernet settings for your module. You must assign an IP address, subnet mask and module address. After you complete this step and download the configuration to the module, you can connect to the module with an Ethernet cable.

1 Determine the network settings for your module, with the help of your network administrator if necessary. You will need the following information:

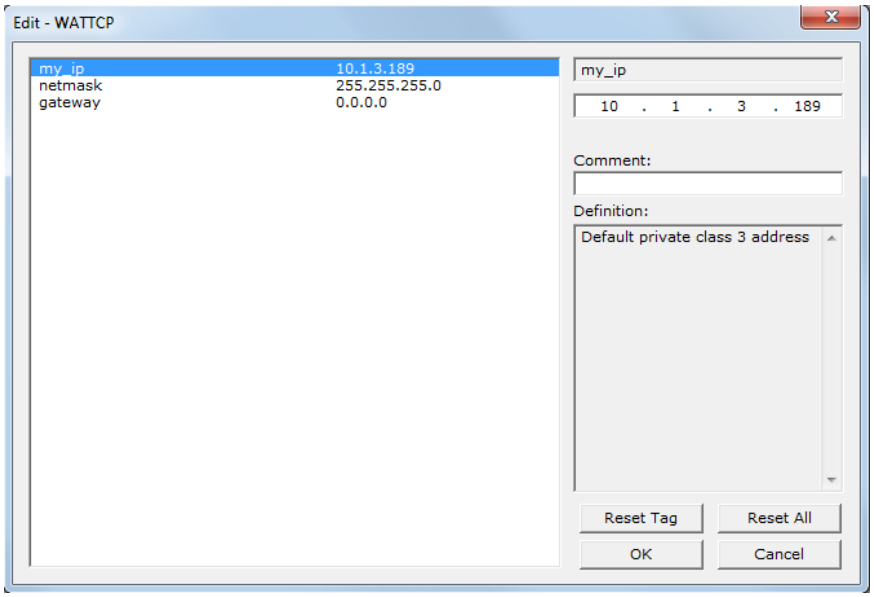
- IP address (fixed IP required) _____ . _____ . _____ . _____
- Subnet mask _____ . _____ . _____ . _____
- Gateway address _____ . _____ . _____ . _____

Note: The module Address is optional, and is not required for networks that do not use a default module.

2 Double-click the **ETHERNET CONFIGURATION** icon.



This action opens the **EDIT** dialog box.



3 Edit the values for my_ip, netmask (subnet mask) and gateway (default gateway).

4 When finished editing, click **OK** to save the changes and return to the ProSoft Configuration Builder window.

2 Configuring the Processor with Unity Pro

In This Chapter

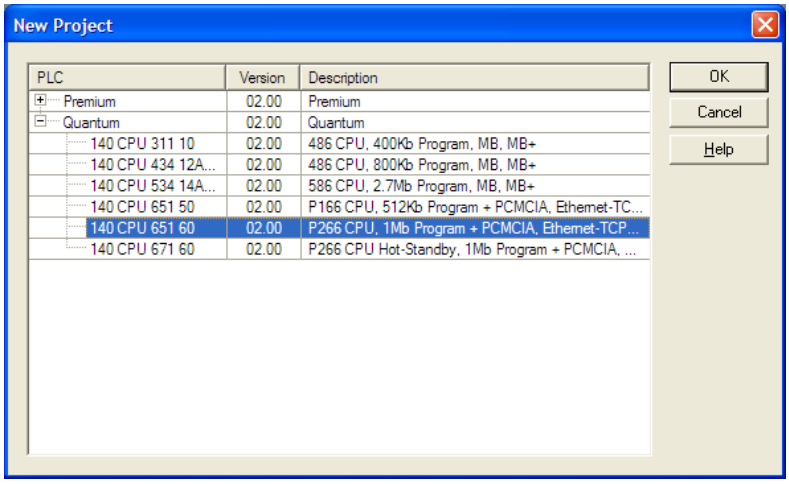
- Create a New Project 22
- Add the PTQ Module to the Project 24
- Build the Project 25
- Connect Your PC to the Processor 26
- Download the Project to the Processor 28

The following steps are designed to ensure that the processor (Quantum or Unity) is able to transfer data successfully with the PTQ module. As part of this procedure, you will use Unity Pro to create a project, add the PTQ module to the project, set up data memory for the project, and then download the project to the processor.

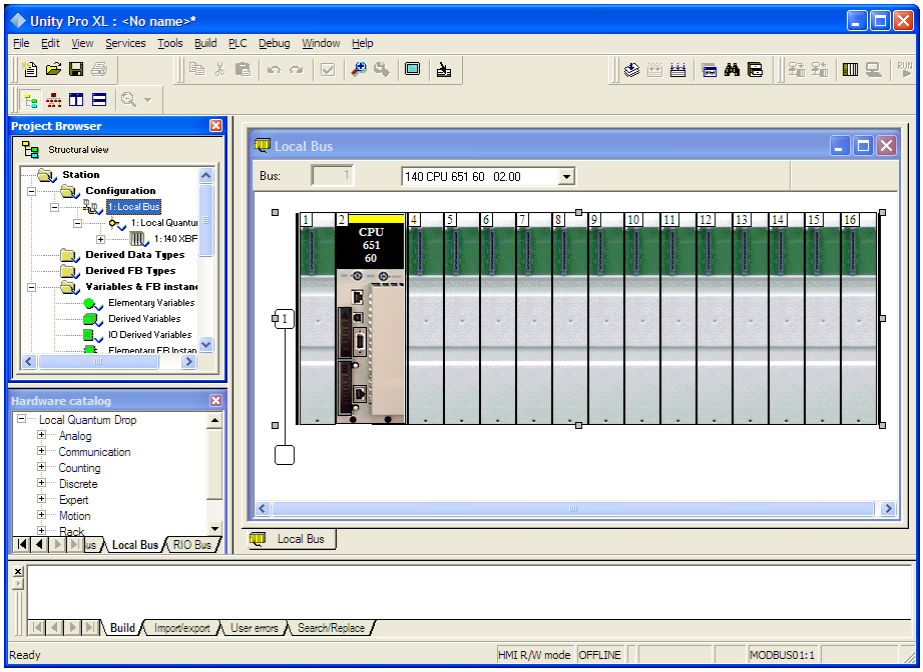
2.1 Create a New Project

The first step is to open Unity Pro and create a new project.

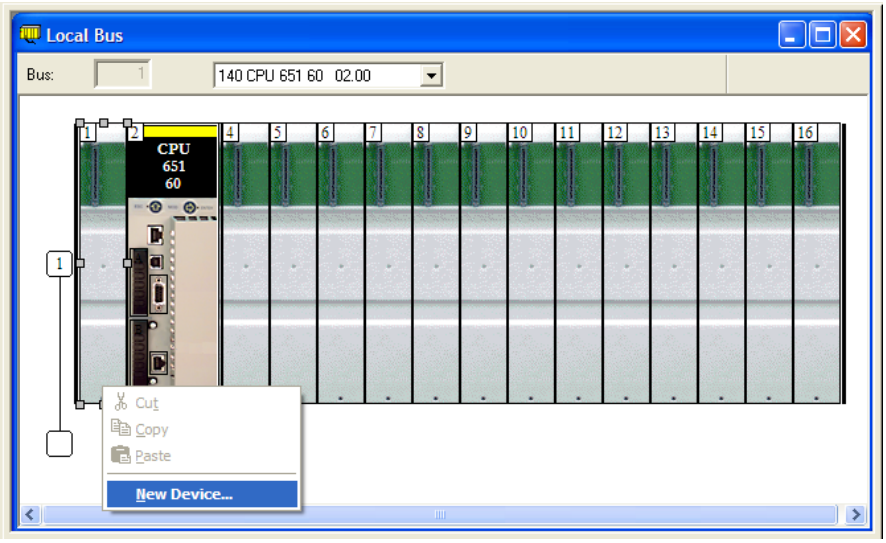
1 In the *New Project* dialog box, choose the CPU type. In the following illustration, the CPU is 140 CPU 651 60. Choose the processor type that matches your own hardware configuration, if it differs from the example. Click **OK** to continue.



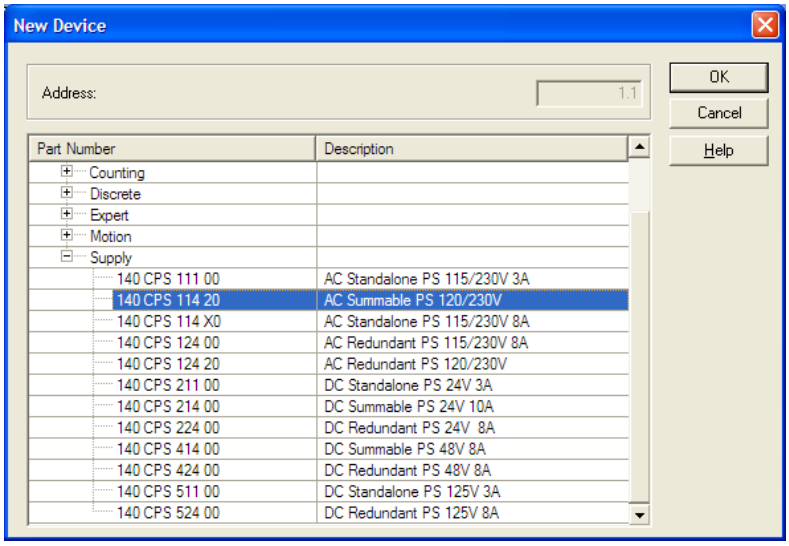
2 Next, add a power supply to the project. In the *Project Browser*, expand the *Configuration* folder, and then double-click the **1:LOCALBUS** icon. This action opens a graphical window showing the arrangement of devices in your Quantum rack.



3 Select the rack position for the power supply, and then click the right mouse button to open a shortcut menu. On the shortcut menu, choose **NEW DEVICE**.



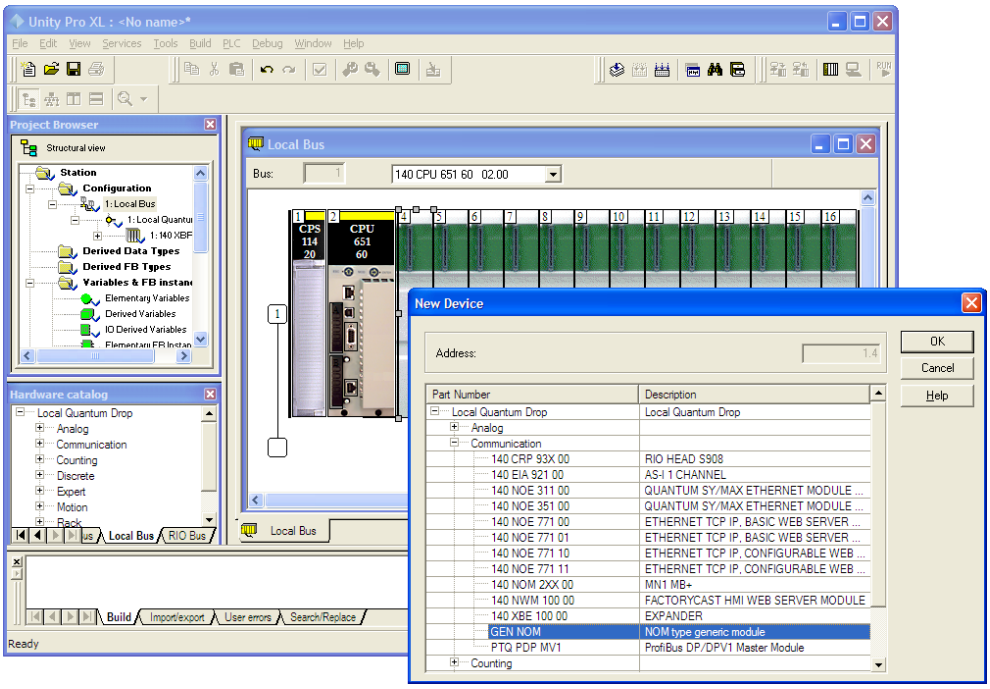
4 Expand the *Supply* folder, and then select your power supply from the list. Click **OK** to continue.



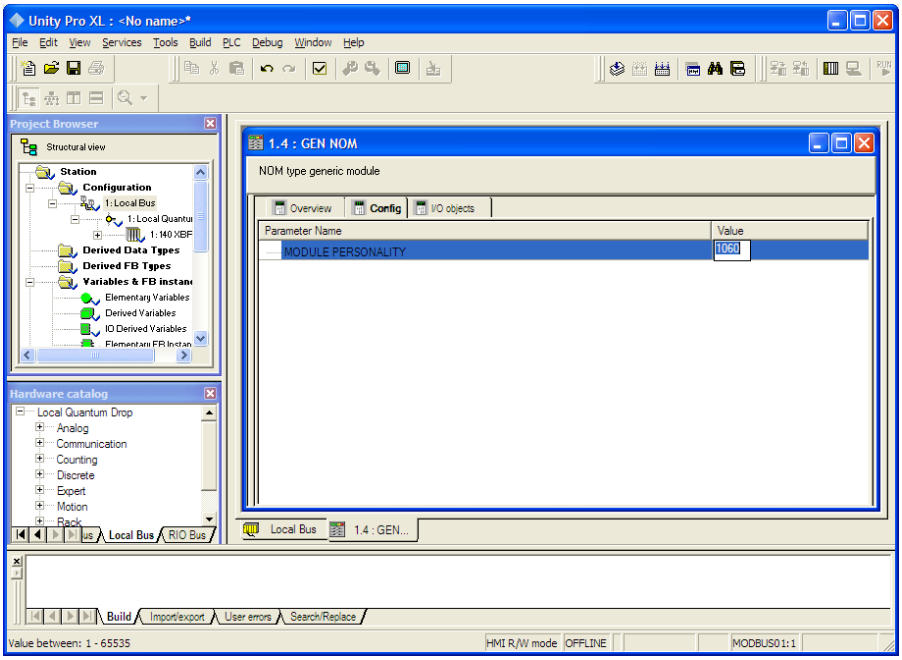
5 Repeat these steps to add any additional devices to your Quantum Rack.

2.2 Add the PTQ Module to the Project

1 Expand the *Communication* tree, and select **GEN NOM**. This module type provides extended communication capabilities for the Quantum system, and allows communication between the PLC and the PTQ module without requiring additional programming.



2 Next, enter the module personality value. The correct value for ProTalk modules is 1060 decimal (0424 hex).



3 Before you can save the project in Unity Pro, you must validate the modifications. Open the **EDIT** menu, and then choose **VALIDATE.** If no errors are reported, you can save the project.
4 **SAVE** the project.

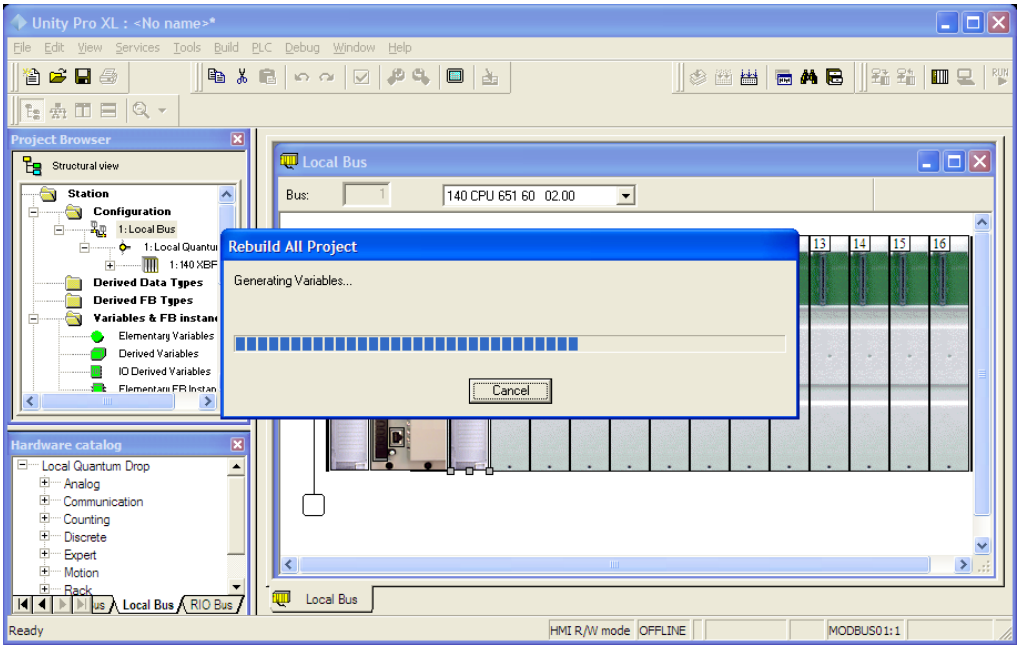
2.3 Build the Project

Whenever you update the configuration of your PTQ module or the processor, you must import the changed configuration from the module, and then build (compile) the project before downloading it to the processor.

Note: The following steps show you how to build the project in Unity Pro. This is not intended to provide detailed information on using Unity Pro, or debugging your programs. Refer to the documentation for your processor and for Unity Pro for specialized information.

To build (compile) the project

1 Review the elements of the project in the *Project Browser*.
2 When you are satisfied that you are ready to download the project, open the **BUILD** menu, and then choose **REBUILD ALL PROJECT**. This action builds (compiles) the project into a form that the processor can use to execute the instructions in the project file. This task may take several minutes, depending on the complexity of the project and the resources available on your PC.
3 As the project is built, Unity Pro reports its process in a *Progress* dialog box, with details appearing in a pane at the bottom of the window. The following illustration shows the build process under way.



After the build process is completed successfully, the next step is to download the compiled project to the processor.

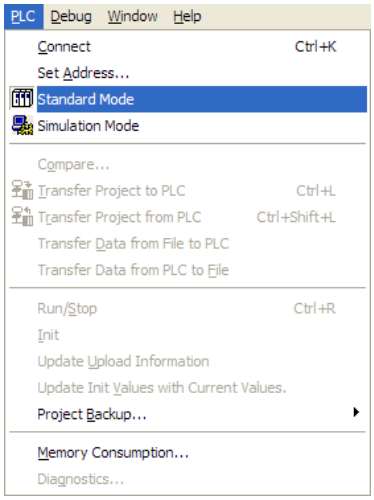
2.4 Connect Your PC to the Processor

The next step is to connect to the processor so that you can download the project file. The processor uses this project file to communicate over the backplane to modules identified in the project file.

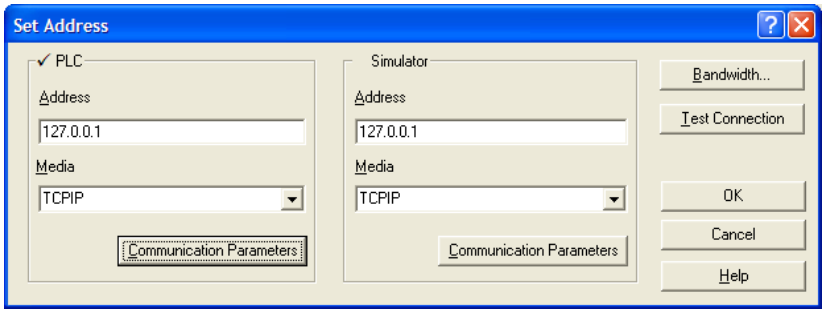
Note: If you have never connected from the PC to your processor before, you must verify that the necessary port drivers are installed and available to Unity Pro.

To verify address and driver settings in Unity Pro

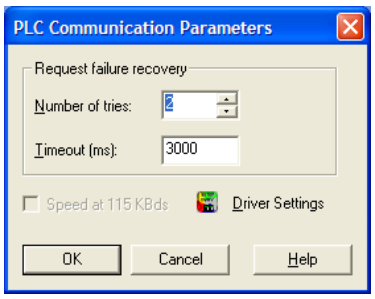
1 Open the **PLC** menu, and choose **STANDARD MODE**. This action turns off the PLC Simulator, and allows you to communicate directly with the Quantum or Unity hardware.



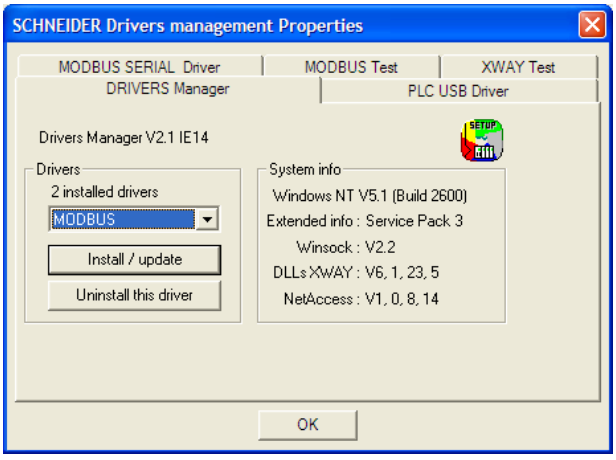
2 Open the **PLC** menu, and choose **SET ADDRESS...** This action opens the *Set Address* dialog box. Open the **MEDIA** dropdown list and choose the connection type to use (TCPIP or USB).



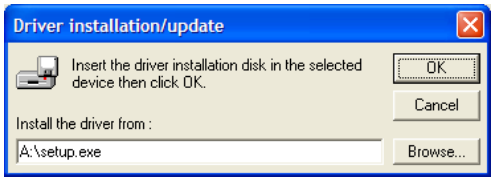
3 If the **MEDIA** dropdown list does not contain the connection method you wish to use, click the **COMMUNICATION PARAMETERS** button in the PLC area of the dialog box. This action opens the *PLC Communication Parameters* dialog box.



4 Click the **DRIVER SETTINGS** button to open the *SCHNEIDER Drivers management Properties* dialog box.



5 Click the **INSTALL/UPDATE** button to specify the location of the Setup.exe file containing the drivers to use. You will need your Unity Pro installation disks for this step.



6 Click the **BROWSE** button to locate the *Setup.exe* file to execute, and then execute the setup program. After the installation, restart your PC if you are prompted to do so. Refer to your Schneider Electric documentation for more information on installing drivers for Unity Pro.

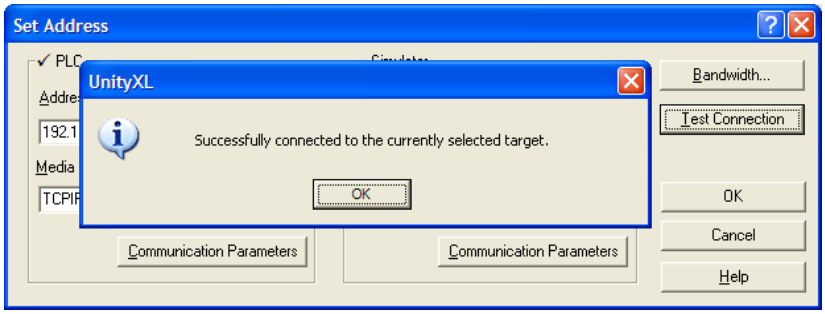
2.4.1 Connecting to the Processor with TCPIP

The next step is to download (copy) the project file to the processor. The following steps demonstrate how to use an Ethernet cable connected from the Processor to your PC through an Ethernet hub or switch. Other connection methods may also be available, depending on the hardware configuration of your processor, and the communication drivers installed in Unity Pro.

1. If you have not already done so, connect your PC and the processor to an Ethernet hub.
2. Open the **PLC** menu, and then choose **SET ADDRESS**.

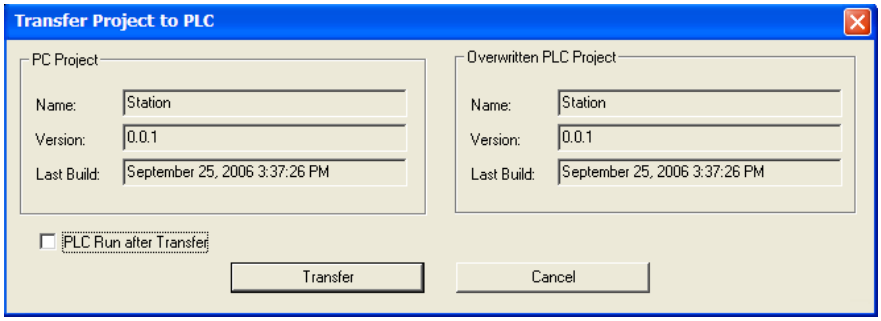
- **Important:** Notice that the *Set Address* dialog box is divided into two areas. Enter the address and media type in the **PLC** area of the dialog box, not the **SIMULATOR** area.

3. Enter the IP address in the address field. In the **MEDIA** dropdown list, choose TCPIP.
4. Click the **TEST CONNECTION** button to verify that your settings are correct.



2.5 Download the Project to the Processor

1. Open the **PLC** menu and then choose **CONNECT.** This action opens a connection between the Unity Pro software and the processor, using the address and media type settings you configured in the previous step.
2. On the **PLC** menu, choose **TRANSFER PROJECT TO PLC**. This action opens the **TRANSFER PROJECT TO PLC** dialog box. If you would like the PLC to go to "Run" mode immediately after the transfer is complete, select (check) the **PLC RUN AFTER TRANSFER** check box.



3 Click the **TRANSFER** button to download the project to the processor. As the project is transferred, Unity Pro reports its process in a **PROGRESS** dialog box, with details appearing in a pane at the bottom of the window.

When the transfer is complete, place the processor in Run mode.

3 Configuring the Processor with Concept

The following steps are designed to ensure the processor is able to transfer data successfully with the PTQ module. As part of this procedure, Concept configuration software from Schneider Electric will be used to create a project, add the PTQ module to the project, set up data memory for the project, and download the project to the processor.

Important Note: Concept software does not report whether the PTQ module is present in the rack Therefore, it is not able to report the health status of the module when the module is online with the Quantum processor. Please consider this when monitoring the status of the PTQ module.

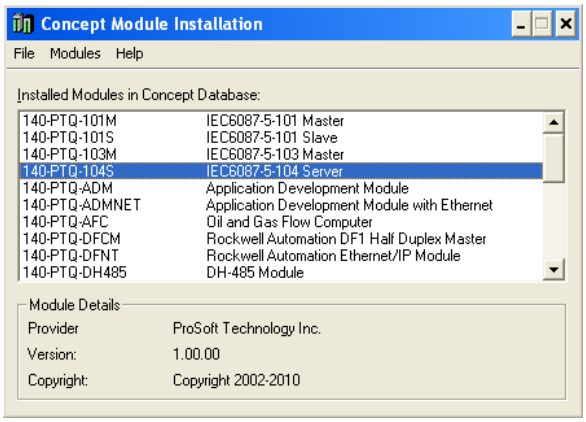
3.1 Information for Concept Version 2.6 Users

This guide uses Concept PLC Programming Software version 2.6 to configure the Quantum PLC. The ProSoft Solutions DVD includes MDC module configuration files that help document the PTQ installation. Although not required, these files should be installed before proceeding to the next section.

3.1.1 Installing MDC Configuration Files

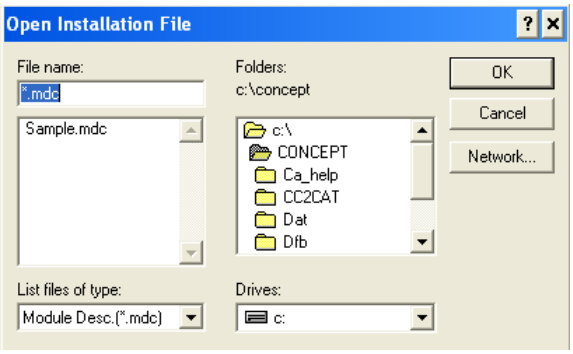
1 From a PC with Concept 2.6 installed, choose **START / PROGRAMS / CONCEPT / MODCONNECT TOOL**.

This action opens the *Concept Module Installation* dialog box.



2 Choose **FILE / OPEN INSTALLATION FILE.**

This action opens the *Open Installation File* dialog box:

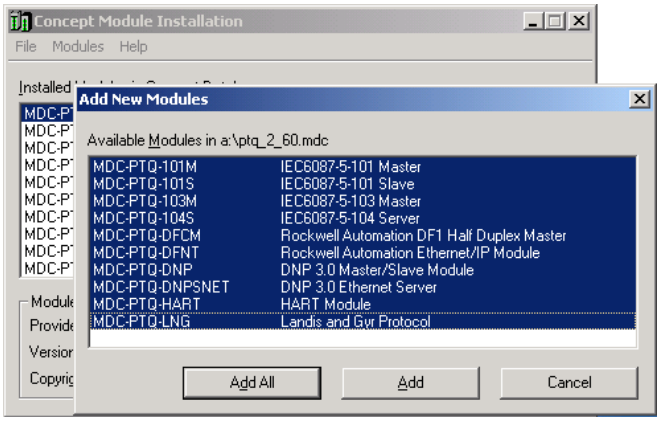


3 If a Quantum processor is being used, the MDC files will be needed. In the *Open Installation File* dialog box, navigate to the *MDC Files* directory on the ProSoft Solutions DVD.

4 Choose the MDC file and help file for the version of Concept:

- Concept 2.6 users: select PTQ_2_60.mdc and PTQMDC.hlp
- Concept 2.5 users: select PTQ_2_50.mdc and PTQMDC.hlp.

Select the files that go with the Concept version being used, and then click **OK**. This action opens the *Add New Modules* dialog box.



5 Click the **ADD ALL** button. A series of message boxes may appear during this process. Click **YES** or **OK** for each message that appears.

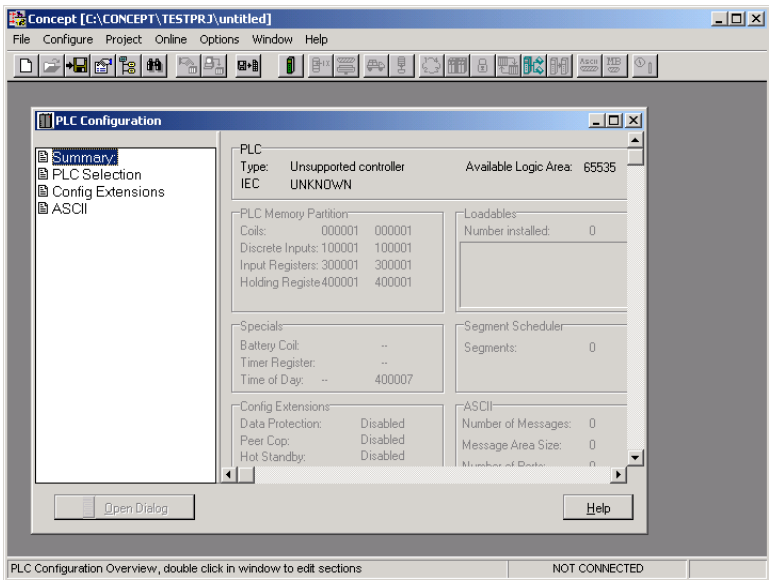
6 When the process is complete, open the **FILE** menu and choose **EXIT** to save the changes.

3.2 Creating a New Project

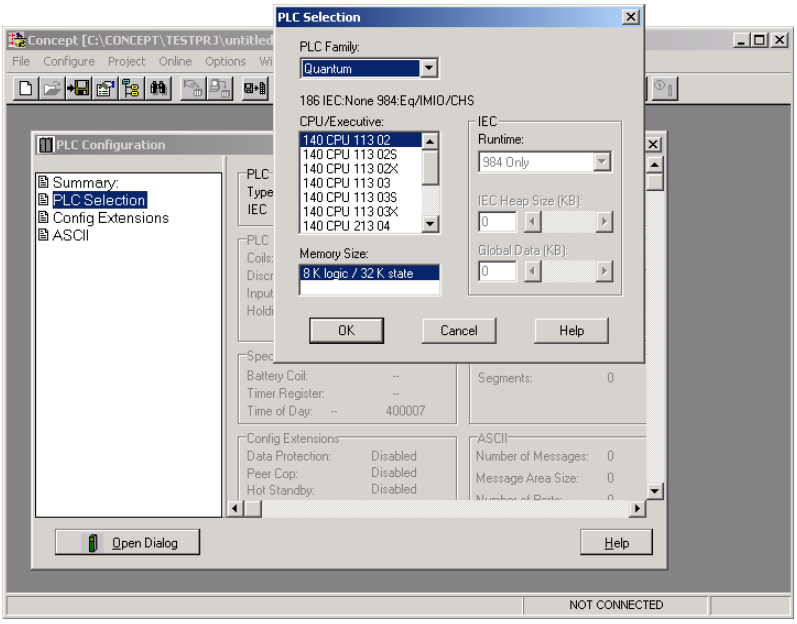
This phase of the setup procedure must be performed on a computer that has the Concept configuration software installed.

1 From the computer, choose **START / PROGRAMS / CONCEPT V2.6 XL.EN / CONCEPT**. This action opens the *Concept* window.

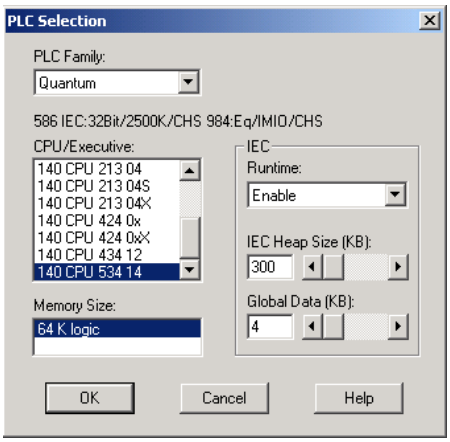
2 Open the File menu, and then choose **NEW PROJECT**. This action opens the *PLC Configuration* dialog box.



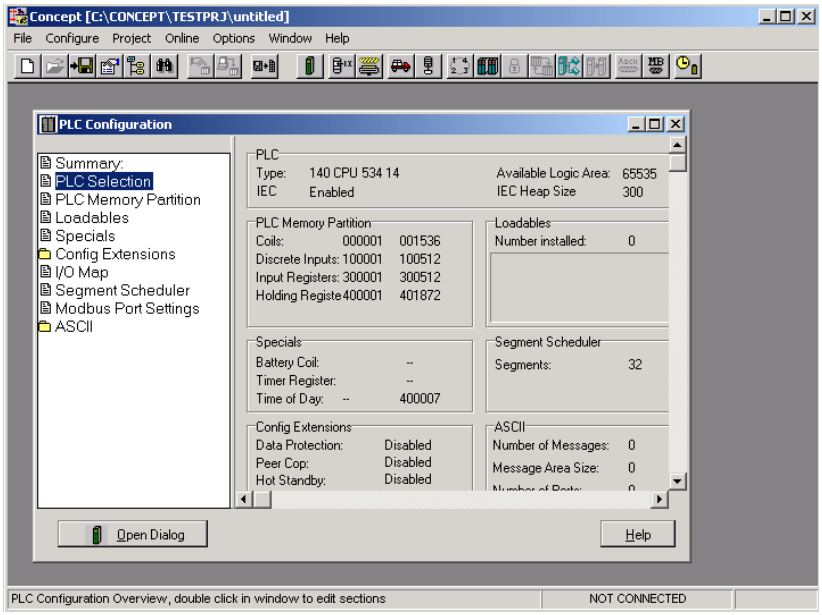
3 In the list of options on the left side of this dialog box, double-click the **PLC SELECTION** folder. This action opens the *PLC Selection* dialog box.



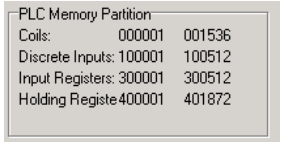
4 In the *CPU/Executive* pane, use the scroll bar to locate and select the **PLC** to configure.



5 Click **OK.** This action opens the *PLC Configuration* dialog box, populated with the correct values that were selected for the PLC.

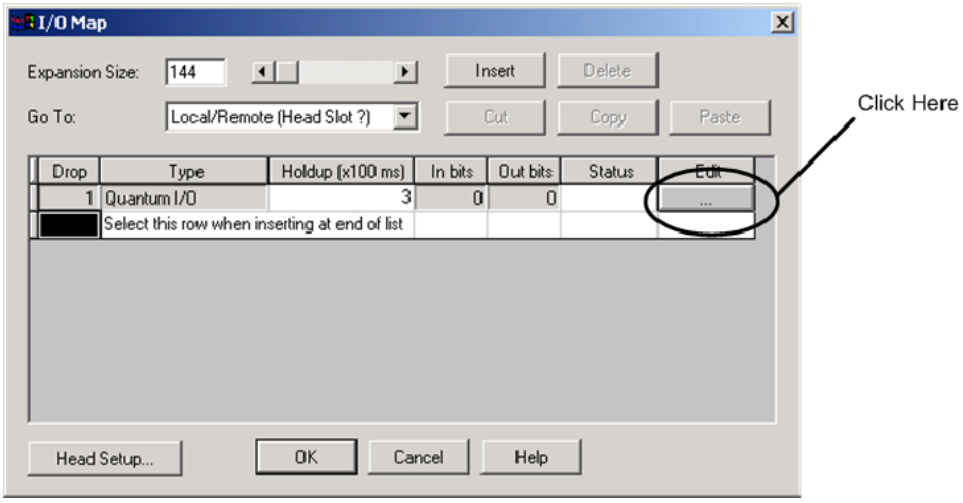


6 Make a note of the holding registers for the module. This information will be needed when modifying the application. The Holding Registers are displayed in the *PLC Memory Partition* pane of the *PLC Configuration* dialog box.

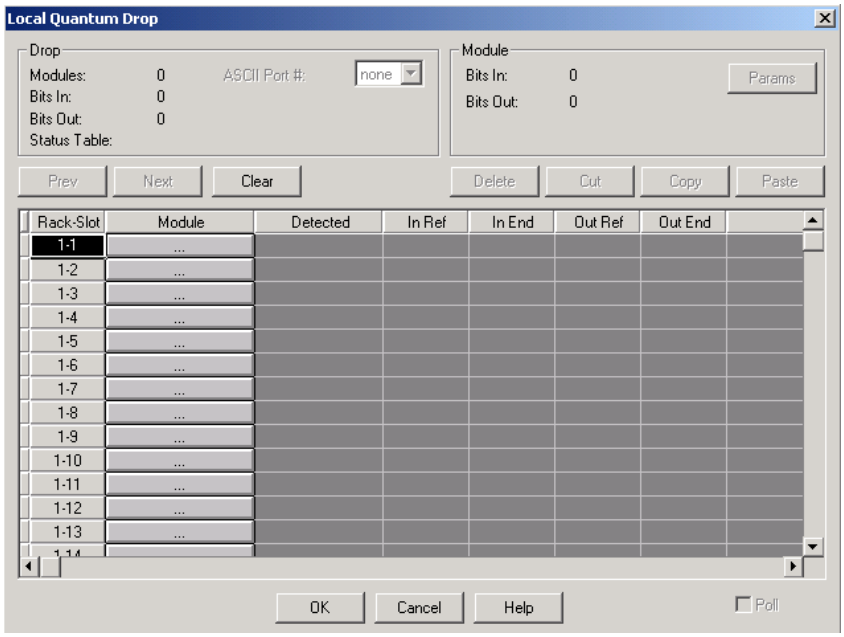


3.3 Adding the PTQ Module to the Project

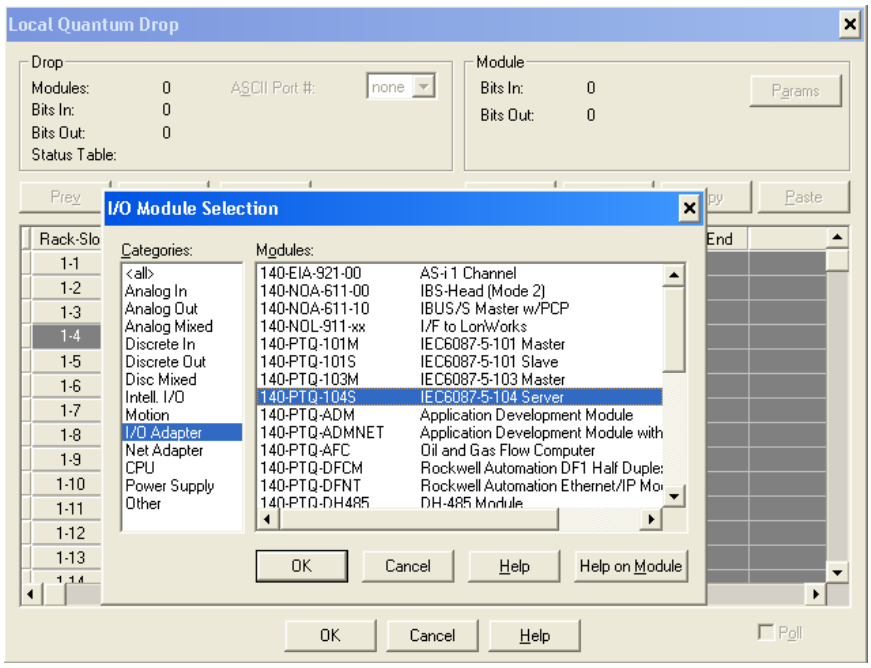
1 In the list of options on the left side of the *PLC Configuration* dialog box, double-click **I/O MAP**. This action opens the *I/O Map* dialog box.



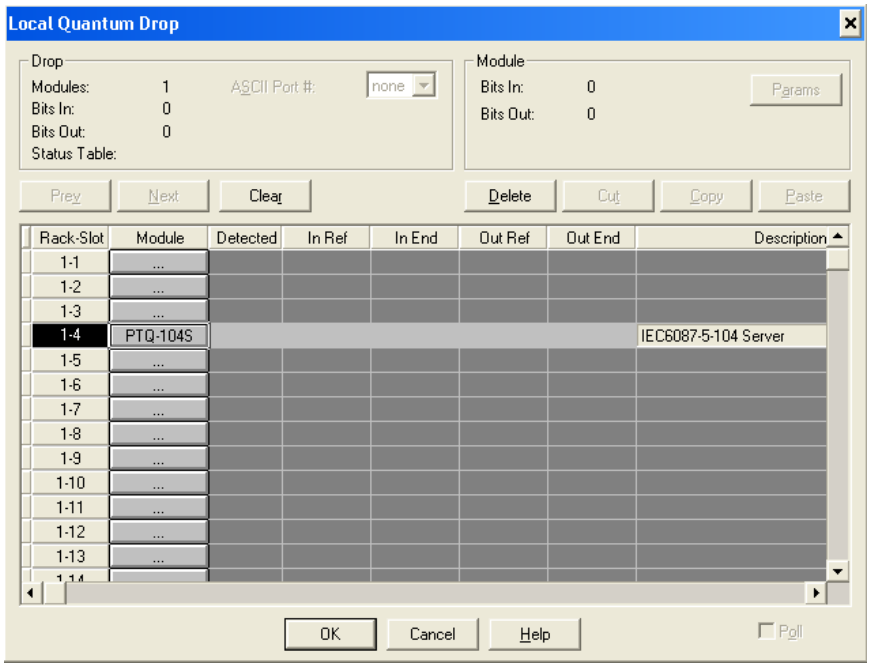
2 Click the **EDIT** button to open the *Local Quantum Drop* dialog box. This dialog box is where rack and slot locations are identified.



3 Click the **MODULE** button next to the rack/slot position where the ProTalk module will be installed. This action opens the *I/O Module Selection* dialog box.



4 In the *Modules* pane, use the scroll bar to locate and select the ProTalk module, and then click **OK.** This action copies the description of the ProTalk module next to the assigned rack and slot number of the *Local Quantum Drop* dialog box.



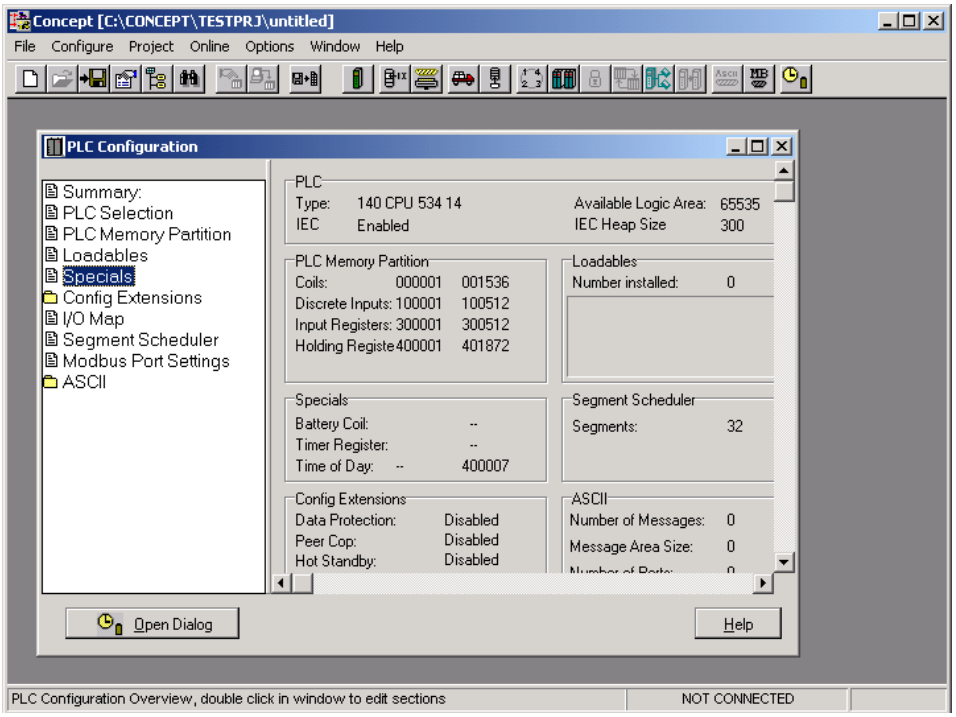
5 Repeat steps 3 through 5 for each ProTalk module that is being installed. When finished installing the ProTalk modules, click **OK** to save the settings. Click **YES** to confirm the settings.

Tip: Select a module, and then click the Help on Module button for help pages.

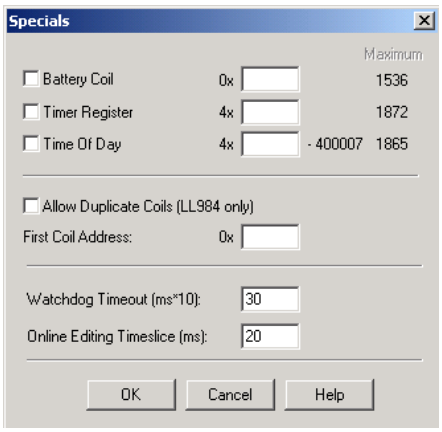


3.4 Setting the Time of Day

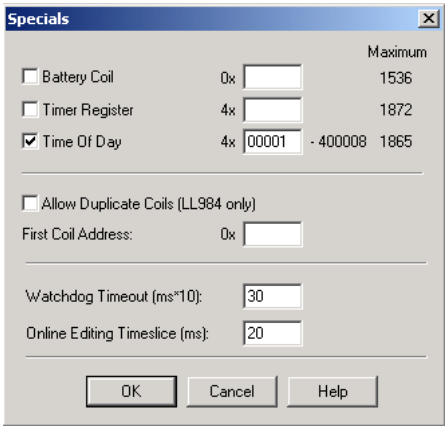
1 In the list of options on the left side of the *PLC Configuration* dialog box, double-click **SPECIALS.**



2 This action opens the **SPECIALS** dialog box.



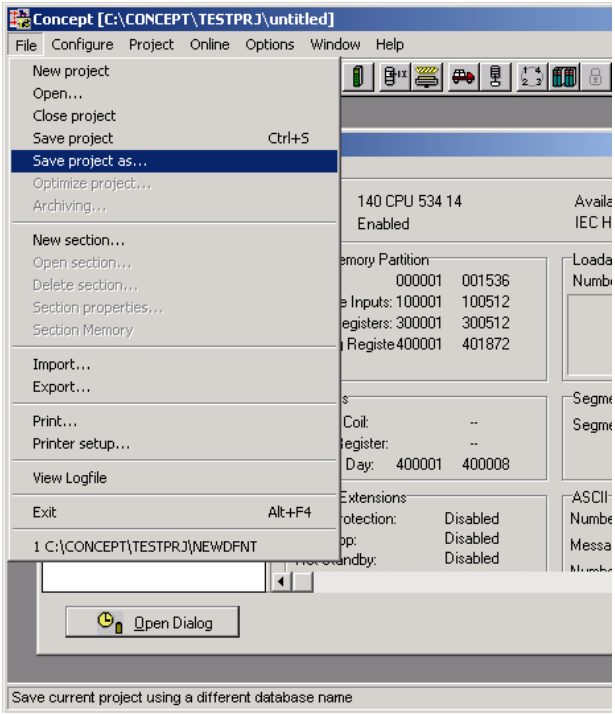
3 Select (check) the *Time of Day* box, and then enter the value 00001 as shown in the following illustration. This value sets the first time of day register to 400001.



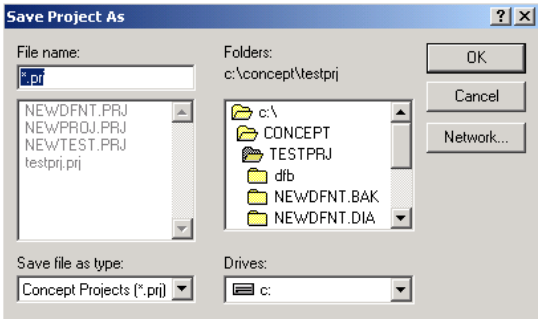
4 Click **OK** to save your settings and close the *Specials* dialog box.

3.5 Saving the Project

1 In the *PLC Configuration* dialog box, choose **FILE / SAVE PROJECT AS.**



2 This action opens the *Save Project As* dialog box.

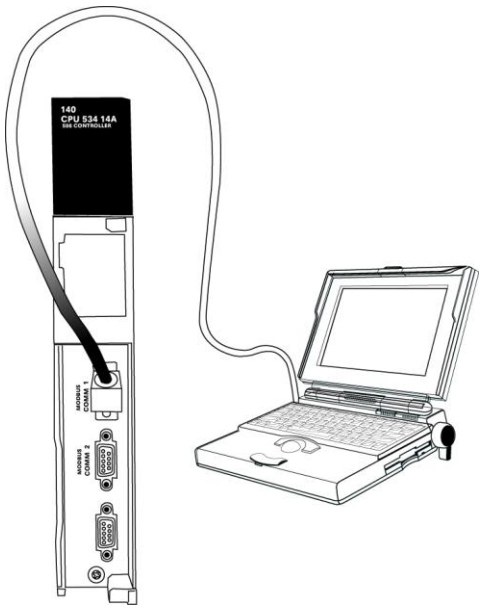


3 Name the project, and then click **OK** to save the project to a file.

3.6 Downloading the Project to the Processor

Next, download (copy) the project file to the Quantum Processor.

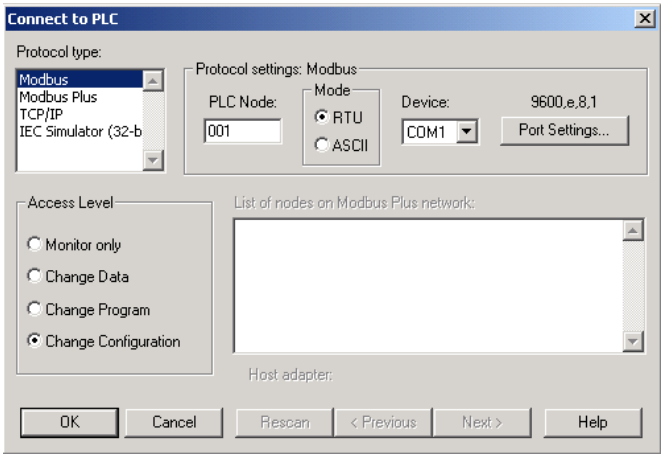
1 Use the null modem cable to connect the PC's serial port to the Quantum processor, as shown in the following illustration.



Note: A Modbus Plus Network Option Module (NOM) can be used in place of the serial port if necessary.

2 Open the **PLC** menu, and then choose **CONNECT.**

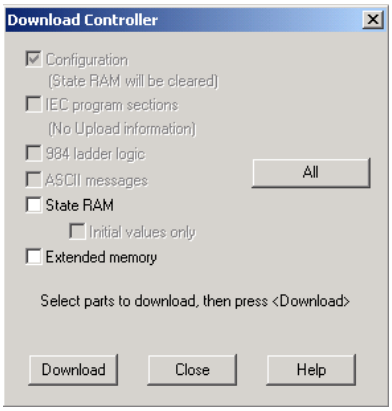
3 In the *PLC Configuration* dialog box, open the **ONLINE** menu, and then choose **CONNECT.** This action opens the *Connect to PLC* dialog box.



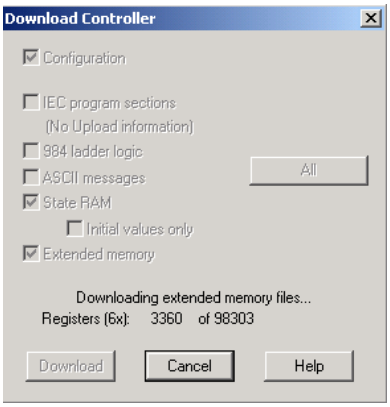
4 Leave the default settings as shown and click **OK.**

Note: Click **OK** to dismiss any message boxes that appear during the connection process.

5 In the *PLC Configuration* window, open the **ONLINE** menu, and then choose **DOWNLOAD.** This action opens the *Download Controller* dialog box.



6 Click **ALL,** and then click **DOWNLOAD.** If a message box appears indicating that the controller is running, click **YES** to shut down the controller. The *Download Controller* dialog box displays the status of the download as shown in the following illustration.



7 When the download is complete, a prompt to restart the controller will display. Click **YES** to restart the controller.

4 Configuring the Processor with ProWORX

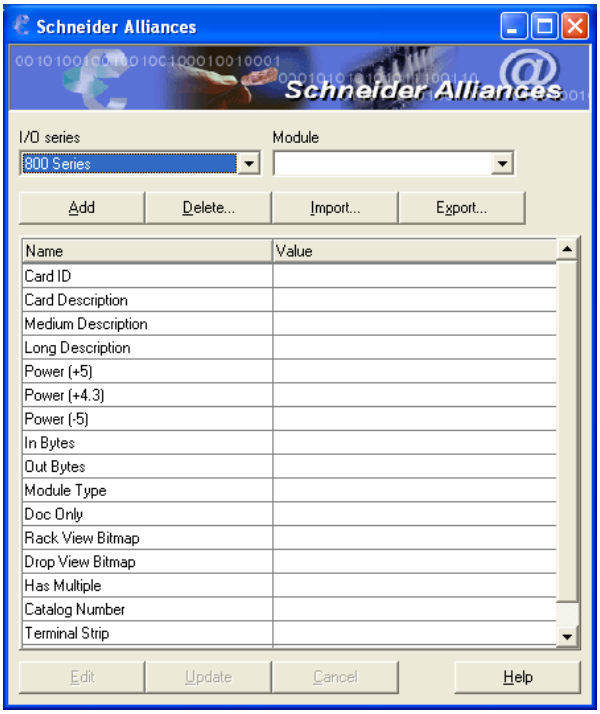
When you use ProWORX 32 software to configure the processor, use the example SAF file provided on the ProTalk Solutions DVD.

Important Note: ProWORX software does not report whether the PTQ module is present in the rack, and therefore is not able to report the health status of the module when the module is online with the Quantum processor. Please consider this when monitoring the status of the PTQ module.

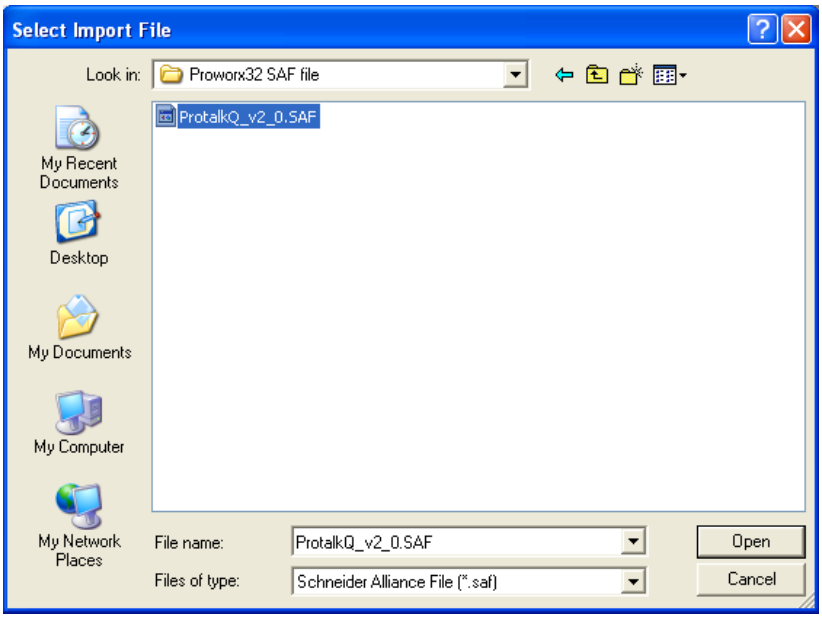
1 Run the **SCHNEIDER_ALLIANCES.EXE** application that is installed with the ProWORX 32 software:



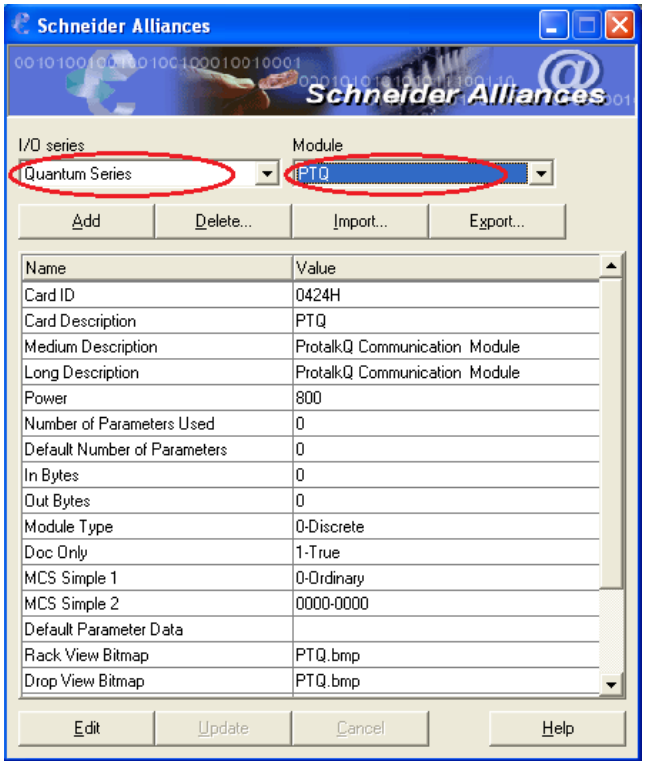
2 Click on **IMPORT…**



3 Select the *.SAF* File that is located on the DVD shipped with the PTQ module.

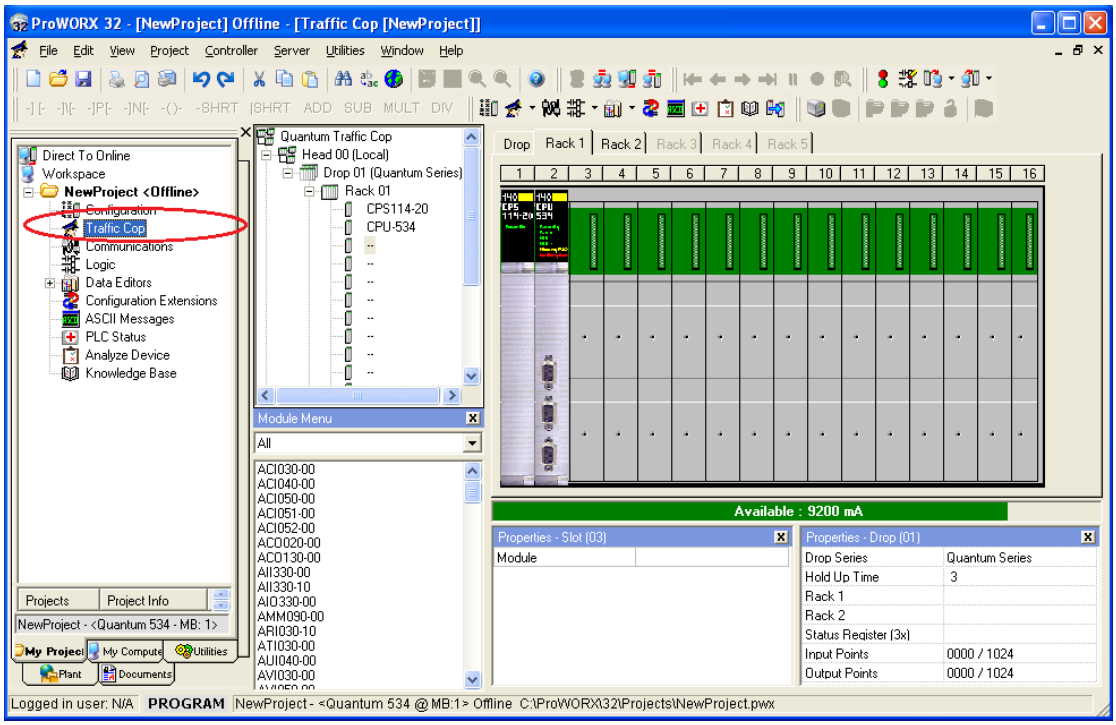


4 After clicking **OPEN,** select the **I/O SERIES** as **QUANTUM SERIES.** Also, select the **MODULE** as **PTQ**:



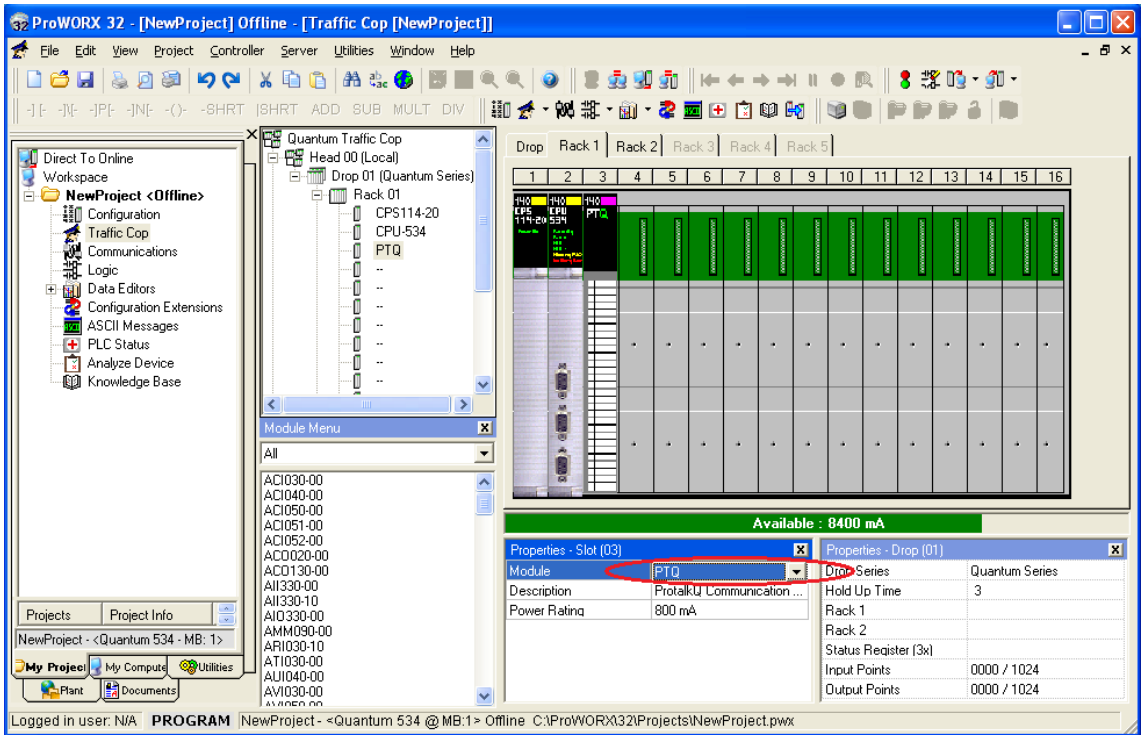
5 Close the Schneider Alliances application.

6 Run the ProWORX 32 software. Whether a new or existing project is used, click on the *Traffic Cop* section to display the rack to be edited.



7 Highlight Slot x by clicking on the slot in the rack display. In this case, the PTQ module will be placed in slot 3.

Below the rack display is the *Properties* drop-down box for slot 3, click and select **PTQ**.



8 Save the project. It is now ready to be downloaded to the Processor.

5 Module Configuration

In This Chapter

- Using ProSoft Configuration Builder 50
- [Backplane Configuration] 54
- Backplane Data Exchange 56
- [SNTP CLIENT] 69
- [IEC-870-5-104] 71
- IEC 60870-5-104 Server section 81
- [IEC-870-5-104 Database] 82
- [IEC-870-5-104 IP Addresses] 86
- [M_SP_NA_1 104] 86
- [M_DP_NA_1 104] 87
- [M_ST_NA_1 104] 87
- [M_BO_NA_1 104] 87
- [M_ME_NA_1 104] 88
- [M_ME_NB_1 104] 88
- [M_ME_NC_1 104] 88
- [M_IT_NA_1 104] 89
- [C_SC_NA_1 104] 89
- [C_DC_NA_1 104] 90
- [C_RC_NA_1 104] 90
- [C_BO_NA_1 104] 90
- [C_SE_NA_1 104] 91
- [C_SE_NB_1 104] 91
- [C_SE_NC_1 104] 92
- Group Codes 92
- CommonNet Data Map 93
- Download the Project to the Module 94
- To Create Optional Comment Entries 93
- To print a configuration file 94

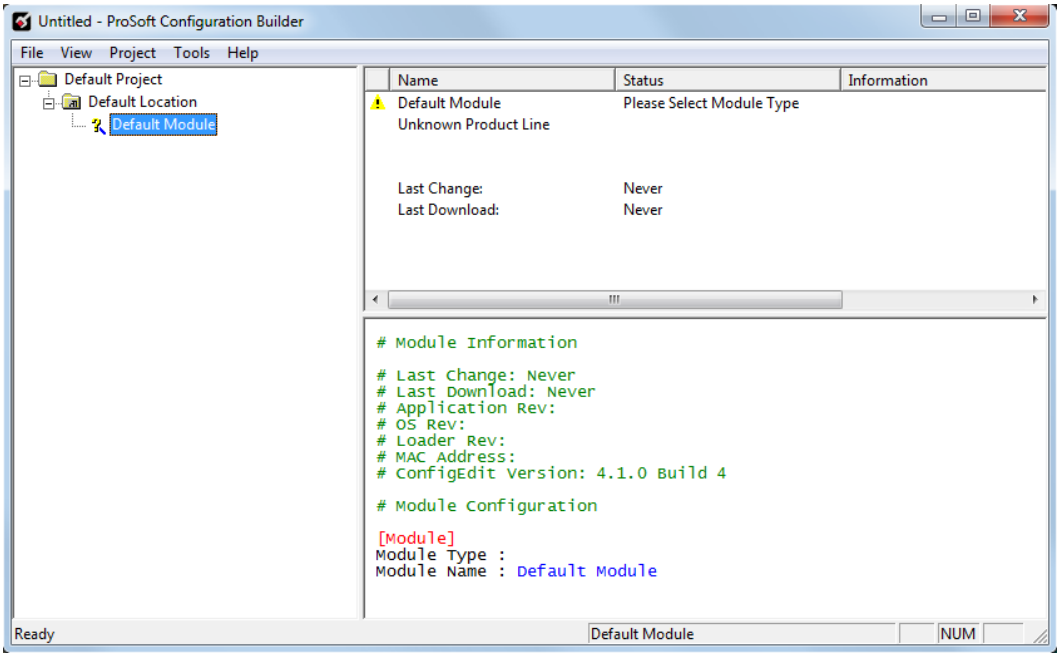
5.1 Using ProSoft Configuration Builder

ProSoft Configuration Builder (PCB) provides a quick and easy way to manage module configuration files customized to meet your application needs. *PCB* is not only a powerful solution for new configuration files, but also allows you to import information from previously installed (known working) configurations to new projects.

5.1.1 Setting Up the Project

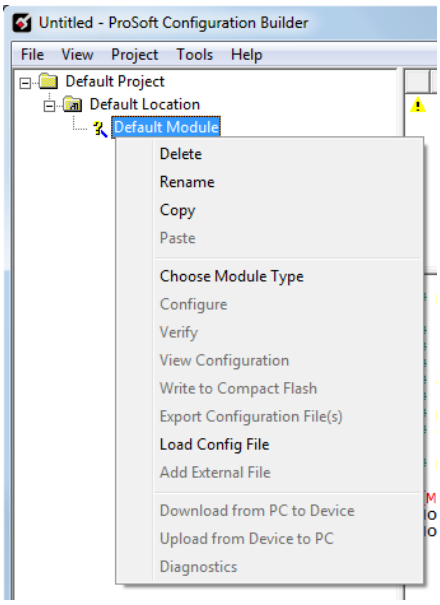
To begin, start ProSoft Configuration Builder. If you have used other Windows configuration tools before, you will find the screen layout familiar. ProSoft Configuration Builder's window consists of a tree view on the left, an information pane and a configuration pane on the right side of the window.

When you first start ProSoft Configuration Builder, the tree view consists of folders for Default Project and Default Location, with a Default Module in the Default Location folder. The following illustration shows the ProSoft Configuration Builder window with a new project.

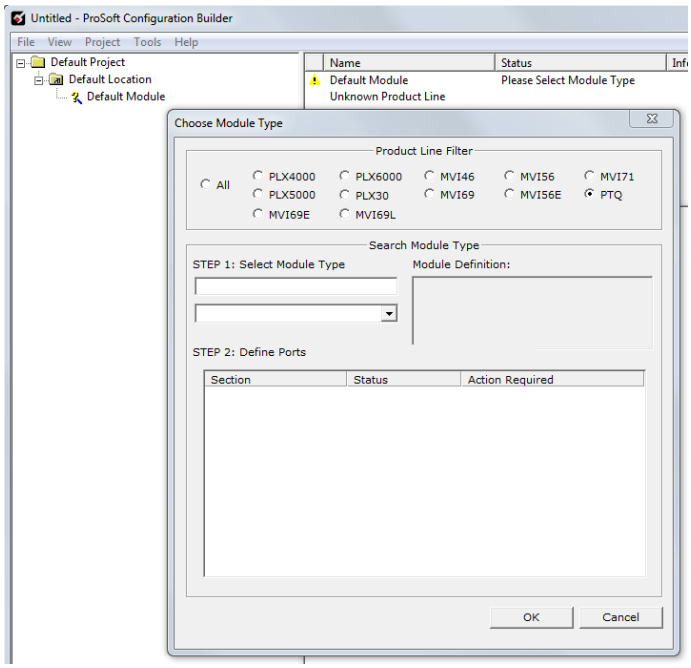


Your first task is to add the PTQ-104S module to the project.

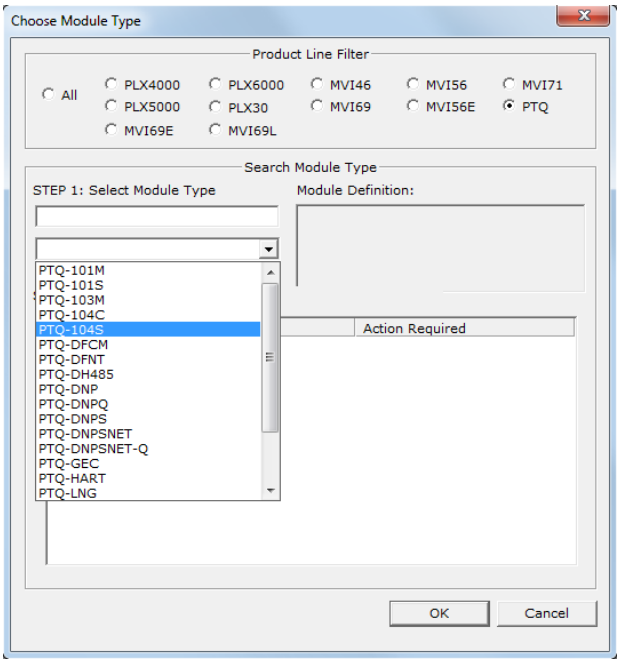
1 Use the mouse to select **DEFAULT MODULE** in the tree view, and then click the right mouse button to open a shortcut menu.



2 On the shortcut menu, choose **CHOOSE MODULE TYPE**. This action opens the **CHOOSE MODULE TYPE** dialog box.



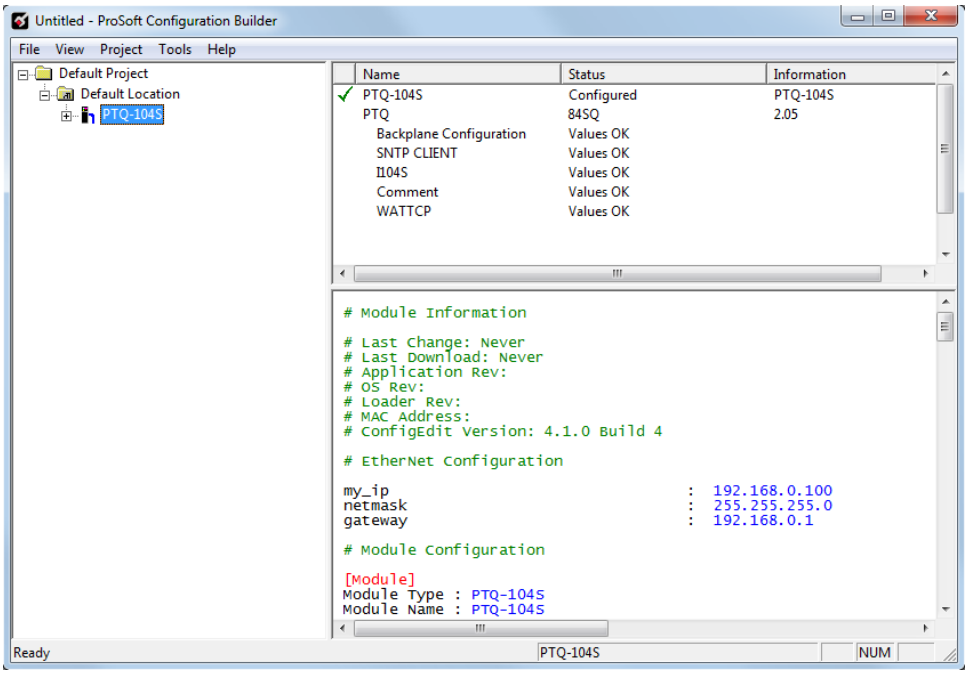
3 In the **PRODUCT LINE FILTER** area of the dialog box, select **PTQ.** In the **SELECT MODULE TYPE** dropdown list, select **PTQ-104S**, and then click **OK** to save your settings and return to the **PROSOFT CONFIGURATION BUILDER** window.



The next task is to set the module parameters.

5.1.2 Set Module Parameters

Notice that the contents of the information pane and the configuration pane changed when you added the PTQ-104S module to the project.



At this time, you may wish to rename the "Default Project" and "Default Location" folders in the tree view.

To rename an object

1. Select the object, and then click the right mouse button to open a shortcut menu. From the shortcut menu, choose **RENAME.**
2. Type the name to assign to the object.
3. Click away from the object to save the new name.

To configure module parameters

1. Click on the plus sign next to the  icon to expand module information.
2. Double-click the  icon to open the *Edit* dialog box.
3. To edit a parameter, select the parameter in the left pane and make your changes in the right pane.
4. Click **OK** to save your changes.

To print a configuration file

1. Select the module icon, and then click the right mouse button to open a shortcut menu.

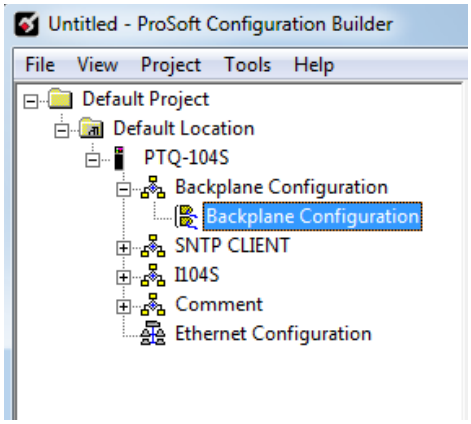
2 On the shortcut menu, choose **VIEW CONFIGURATION.** This action opens the *View Configuration* window.
3 On the *View Configuration* window, open the **FILE** menu, and choose **PRINT.** This action opens the *Print* dialog box.
4 On the *Print* dialog box, choose the printer to use from the dropdown list, select printing options, and then click **OK.**

5.2 [Backplane Configuration]

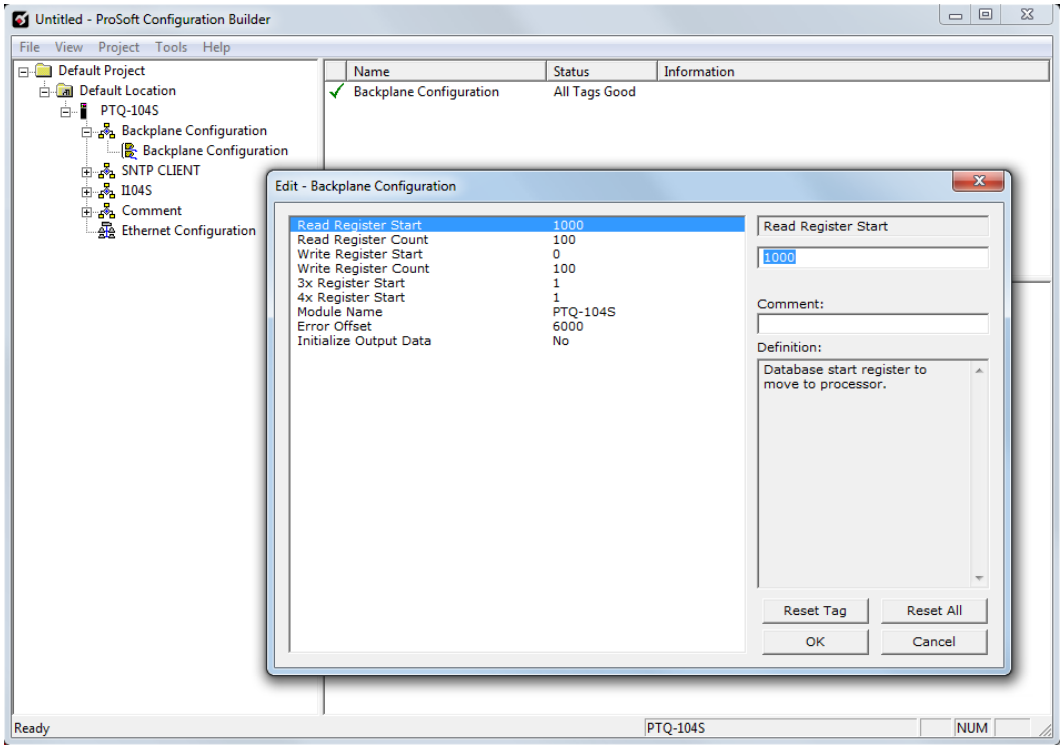
On every scan, the PTQ-104S transfers all Read and Write database values between the module and the processor.

The module will hold the processor scan for a certain period of time to complete the transfer. Therefore, the larger the read and write areas, the longer the processor scan time will be.

The [Backplane Configuration] section of PCB defines the starting registers for read and write operations. It also defines the length of each data area.



Double click the [Module] icon to edit.



5.2.1 Read Register Start

0 through 3999
Database start register to move to processor

5.2.2 Read Register Count

0 through 3999
Number of words moved from module to processor

5.2.3 Write Register Start

0 through 3999
Database start register where data placed from processor

5.2.4 Write Register Count

0 through 3999
Number of words moved from processor to module

5.2.5 3X Register Start

1 through 65535

3x start register where data moved from module to processor

5.2.6 4X Register Start

1 through 65535

4x start register where data moved from processor to module

5.2.7 Module Name

Name to display on module

5.2.8 Error Offset

1 through 65535

Internal register of module to store error and status – recommended within the read data range of module memory. Example: 1000-1099 according to screenshot above.

5.2.9 Initialize Output Data

Yes or No

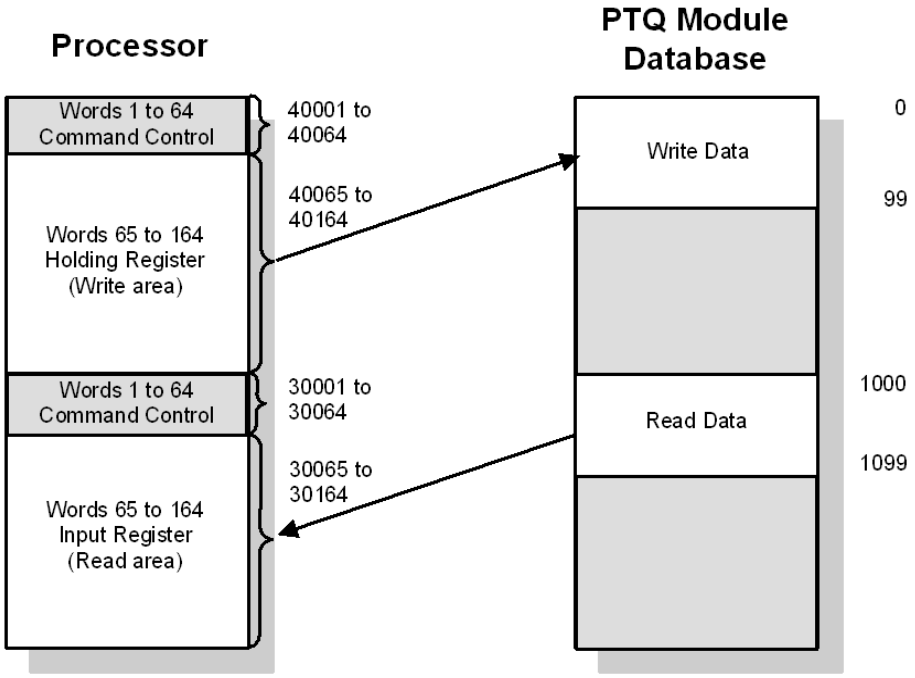
This parameter determines if the output data for the module should be initialized with values from the processor. If the value is set to No (0), the output data will be initialized to 0. If the value is set to Yes (1), the data will be initialized with data from the processor. Use of this option requires associated ladder logic to pass the data from the processor to the module.

5.3 Backplane Data Exchange

The module transfers all Read and Write data between the module and the processor on every scan. Up to 4000 total words can be configured in the module database. The more data that is configured, the longer the processor scan will take.

The values in the example PCB configuration file from the previous section are illustrated in the following diagram.

Read Register Start: 1000
Read Register Count: 100
Write Register Start: 0
Write Register Count: 100



Note: The diagram above shows the memory addresses for a Quantum processor. If using a Unity processor, substitute %MW for read only data, and %IW for read/write data.

Words 0 through 63 in each read/write block are reserved for special functions. Refer to Command Control Blocks (Page 61) for more information on special function blocks. The following table shows the relationship between the processor memory and the module database areas.

Module Database	Register	Unity Register	Description
Read Data	3x	%IW	Input Register
Write Data	4x	%MW	Holding Register

The data mapping in the following example shows the relationship between processor and PTQ-104S memory addresses, assuming a 4x register start value of 40001 and a PTQ-104S database start value of 0.

Processor Memory Address	Module Database Address
40065	0
40066	1
40067	2
40068	3
40069	4
…	…
40164	99

The data mapping in the following example shows the relationship between processor and PTQ-104S memory addresses, assuming a 3x register start value of 30001 and a PTQ-104S database start value of 2000.

Processor Memory Address	Module Database Address
30065	2000
30066	2001
30067	2002
30068	2003
30069	2004
…	…
30164	2099

5.3.1 Data Type Mapping and Addressing

When interfacing data in the processor to that of the IEC 60870-5-104 protocol, it is important the user understands the mapping of the data types to their corresponding representation in the modules database. The table that follows lists the data types supported by the module and their associated storage representation.

IEC-870-5-104 Data Types

Type ID	Type	Description	Data representation
1	M_SP_NA_1 (7.3.1.1)	Monitored Single-point Information: This data type stores a single binary input point. Associated time-tagged event information for this type are M_SP_TA_1 (2) and M_SP_TB_1 (30).	Single bit value (7.2.6.1) with 0=Off and 1=On.
3	M_DP_NA_1 (7.3.1.3)	Monitored Dual-point Information: This data type stores a dual-point binary input value (that is, valve status). Associated time-tagged event information for this type are M_DP_TA_1 (4) and M_DP_TB_1 (31).	Dual-bit status (7.2.6.2) with 00b (0 decimal) = indeterminate or intermediate, 01b (1 decimal) = Off, 10b (2 decimal) = On and 11b (3 decimal) = indeterminate.
5	M_ST_NA_1 (7.3.1.5)	Monitored Step-point Information: This data type is used for step position of transformers or other step position information. The value for the position ranges from -64 to 63. Associated time-tagged event information for this type are M_ST_TA_1 (6) and M_ST_TB_1 (32).	Step data (7.2.6.5) is stored in a single character value with bits 0 to 6 (-64 to +63) representing the step position and bit 7 representing the following states: 0 = Equipment is not in transient state 1 = Equipment in transient state

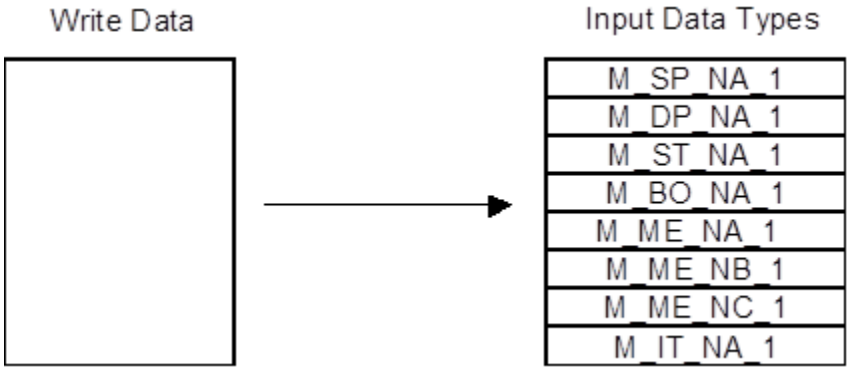
Type ID	Type	Description	Data representation
7	M_BO_NA_1 (7.3.1.7)	Monitored Bitstring of 32-bit data --This data type stores 32-bit data in binary form. Each bit in the string has a value of 0 or 1. Associated time-tagged event information for this type are M_BO_TA_1 (8) and M_BO_TB_1 (33).	Each of the 32 bits in the bitstring has a value of 0 or 1 (7.2.6.13).
9	M_ME_NA_1 (7.3.1.9)	Monitored Normalized Measured Value: This data type is used for analog input data. Associated time-tagged event information for this type are M_ME_TA_1 (10) and M_ME_TD_1 (34).	Normalized values (7.2.6.6) are stored in a word (16-bit) data area with a range of -1..+1-2^{-15}
11	M_ME_NB_1 (7.3.1.11)	Monitored Scaled Measured Value --This data type is used for analog input data. Associated time-tagged event information for this type are M_ME_TB_1 (12) and M_ME_TE_1 (35).	Scaled values (7.2.6.7) are stored in a word (16-bit) data area with a range of -2^{15}.. +2^{15}-1
13	M_ME_NC_1 (7.3.1.13)	Monitored Measured Value, Short Floating-Point Number: This data type is used for analog input data stored in floating point format according to the IEEE STD 754, QDS format. Associated time-tagged event information for this type are M_ME_TC_1 (14) and M_ME_TE_1 (36).	Short floating-point number stored in IEEE STD 754 format (Fraction, Exponent, Sign) (7.2.6.8)
15	M_IT_NA_1 (7.3.1.15)	Monitored Integrated Total-point Information -- This data type stores meter or other count data. Associated time-tagged event information for this type are M_IT_TA_1 (15) and M_IT_TB_1 (37).	Binary counter data (7.2.6.9) is stored in a double-word (32-bit) value with a range of -2^{31}..+2^{31}-1.
45	C_SC_NA_1 (7.3.2.1)	Single-point Command: This command controls a single binary point such as a relay.	Single bit value (7.2.6.15) with 0 = Off and 1 = On
46	C_DC_NA_1 (7.3.2.2)	Double-point Command: This command controls a dual-point binary control device such as a trip/close relay.	Double Command (7.2.6.16) with 0 = Not permitted 1 = Off 2 = On 3 = Not permitted
47	C_RC_NA_1 (7.3.2.3)	Regulating Step Command: This command controls a stepping device such as a transformer.	Regulating Step Command (7.2.6.17) with 0 = Not permitted 1 = Next step lower 2 = Next step higher 3 = Not permitted
48	C_SE_NA_1 (7.3.2.4)	Setpoint Command, Normalized Value: This command controls an analog device.	Normalized values (7.2.6.6) are stored in a word (16-bit)data area with a range of -1..+1-2^{-15}
49	C_SE_NB_1 (7.3.2.5)	Setpoint Command, Scaled Value: This command controls an analog device.	Scaled values (7.2.6.7) are stored in a word (16-bit) data area with a range of -2^{15}.. +2^{15}-1
50	C_SE_NC_1 (7.3.2.6)	Setpoint Command, Short Floating-Point Format: This command controls an analog device accepting an IEEE STD 754 floating-point format value.	Short floating-point number stored in IEEE STD 754 format (Fraction, Exponent, Sign) (7.2.6.8)
51	C_BO_NA_1 (7.3.2.7)	Setpoint Command, 32-bit Bitstring: This command controls a bitstring in a device.	Each of the 32 bits in the bitstring has a value of 0 or 1 (7.2.6.13).

The data addressing is resumed in the following table.

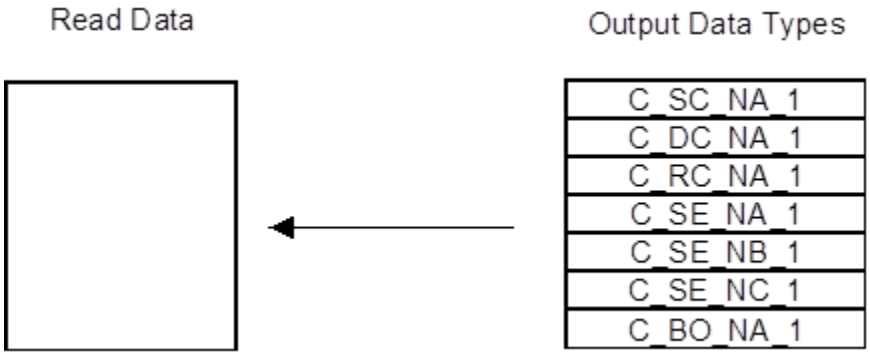
Data	Size	Example
Single Point	1 bit	Address 1600 refers to word 100, bit 1 in database
Dual Point	2 bits	Address 1600 refers to word 100, bits 1 and 2 in database
Step Point	1 byte	Address 200 refers to word 100, lower byte in database
Bitstring 32 bit	2 words	Address 50 refers to word 100 and 101 in database

Data	Size	Example
Normalized Measured Value	1 word	Address 100 refers to word 100 in database
Scaled Measured Value	1 word	Address 100 refers to word 100 in database
Short Float Point Measured Value	2 words	Address 50 refers to words 100 and 101 in database
Integrated Total Point	2 words	Address 50 refers to words 100 and 101 in database

Another important concept to understand is the direction of data transfer for the different data types with reference to the controller. The following illustration shows the data types (monitored data) that are transferred from the processor to the module. This monitor data can be read on the IEC-870-5-104 network.



The next diagram shows the movement of control data from the module to the processor.



As blocks are transferred between the module and the processor, each block contains block identification codes that define the content or function of the block of data transferred. The control blocks used by the module are displayed in the following table.

Block Range	Descriptions
9250	Status Data
9958	Event messages
9970	Set PLC time using module's time
9971	Set module's time using PLC time
9998	Warm Boot Request from PLC (Block contains no data)
9999	Cold Boot Request from PLC (Block contains no data)

Block identification codes 9901 to 9999 are used for special control blocks to control the module. Each of these blocks is discussed in the following topics.

Normal Data Transfer Blocks

These data are transferred through read (input image) and write (output image) blocks. Refer to Module Configuration for a description of the data objects used with the blocks and the ladder logic required. The following topics discuss the structure and function of each block.

Input Data (3x Register Data)

These blocks of data transfer information from the module to the Quantum processor. The following table describes the structure of the input image.

Offset	Description	Length
0	Sequence Counter	1
1	Block ID	1
2 to 63	Command Response Data	62
64 to n	Read Data	0 to 3999

Output Data (4x Register Data)

These blocks of data transfer information from the Quantum processor to the module. The following table describes the structure of the output image.

Offset	Description	Length
0	Sequence Counter	1
1	Block ID	1
2 to 63	Command Data	62
64 to n	Write Data	0 to 3999

Command Control Blocks

Block identification codes greater than 9900 are utilized to perform special functions in the module. Each control block recognized and used by the module is defined in the following topics.

Command Codes	Descriptions
9250	Status Block
9958	Event messages
9970	Set PLC time using module's time
9971	Set module's time using PLC time
9998	Warm Boot Request from PLC (Block contains no data)
9999	Cold Boot Request from PLC (Block contains no data)

Note: The command code in the I/O area is also referred to as the block ID.

Implementing Ladder to Support Special Functions

In order to use Special Functions, some form of control logic must be implemented. The following section uses structured text language to illustrate how a typical function might be implemented.

Example: Rebooting the Module.

MyTrigger is a variable that triggers this logic

*OutputContro*l variable array starts at register 4000001

The first instruction guarantees that the processor requests this block for only one scan.

The second instructions sets the Block Number (9999 = ColdBoot) and then sets the sequence number to 1.

```
IF MyTrigger>0 AND OutputControl1[1]> 0 THEN
 OutputControl1[0]:= InputData[0];
 OutputControl1[1]:=0;
  MyTrigger :=0;
END_IF;

IF (MyTrigger=9999)OR (MyTrigger=9998) OR (MyTrigger=9250) THEN
 OutputControl1[1] :=MyTrigger;
 Temp:=WORD_TO_INT(OutputControl1[0]);
 Temp:=Temp+1;
 OutputControl1[0]:=INT_TO_WORD(Temp);

 END_IF;
```

Example: Retrieving the time of day from the module.

This logic shows an example on how to request a block 9970 from the module (Read Module's Time) and read the response to the processor.

Assumptions:

- MyTrigger is a variable that triggers this logic
- OutputControl variable array starts at register 400,001
- InputControl variable array starts at register 300,001
- MyTime variables store the date and time values to be read from the module

Sets the Block Number (9970=Read Module's Time) and then increments the output sequence number (OutputControl[1]) by one. Once the module reads a new output sequence number from the processor it will process this request. So remember that the actual trigger is moving a new output block sequence number value to the module. Moving the block number (9970) is not the trigger to request this task from the module.

MyTrigger is set to -1 as an indication that the logic is waiting for the response from the module.

```
IF (MyTrigger=9970) THEN
 OutputControl1[1] :=MyTrigger;
 Temp:=WORD_TO_INT(OutputControl1[0]);
 Temp:=Temp+1;
 OutputControl1[0]:=INT_TO_WORD(Temp);

END_IF;
```

When the request is processed, the module will send the block response and increment the received output sequence number by 1. So the output sequence number is one less than the input sequence number the module has sent a new block. Once the block is received the processor logic copies the received data to the appropriate variables. The logic also clears the trigger for the next request.*)

```
IF (InputData[1]=9970) THEN
GetTime.MyYear :=InputData[2];
GetTime.MyMonth :=InputData[3];
GetTime.MyDay :=InputData[4];
GetTime.MyHour :=InputData[5];
GetTime.MyMinute :=InputData[6];
GetTime.MySecond :=InputData[7];
GetTime.MyMillisecond :=InputData[8];
END_IF
```

Example: Setting the time of day to the module.

This logic shows an example on how to request a block 9971 from the module (Read Module's Time).

Assumptions:

- MyTrigger is a variable that triggers this logic
- OutputControl variable array starts at register 400,001
- InputControl variable array starts at register 300,001
- MyTime variables store the date and time values to be written to the module

Sets the Block Number (9971=Write Module's Time) and then increments the output sequence number (OutputControl[1]) by one. Once the module reads a new output sequence number from the processor it will process this request. So remember that the actual trigger is moving a new output block sequence number value to the module. Moving the block number (9970) is not the trigger to request this task from the module. MyTrigger is set to -1 as an indication that the logic is waiting for the response from the module.

```
IF (MyTrigger=9971) THEN
OutputControl1[1] :=MyTrigger;
OutputControl1[2] :=SetTime.MyYear;
OutputControl1[3] :=SetTime.MyMonth;
OutputControl1[4] :=SetTime.MyDay;
OutputControl1[5] :=SetTime.MyHour;
OutputControl1[6] :=SetTime.MyMinute;
OutputControl1[7] :=SetTime.MySecond;
OutputControl1[8] :=SetTime.MyMillisecond;
Temp:=WORD_TO_INT(OutputControl1[0]);
Temp:=Temp+1;
OutputControl1[0]:=INT_TO_WORD(Temp);

END_IF;
```

Once the request was processed the module will send the block response and increment the received output sequence number by 1. So the output sequence number is one less than the input sequence number the module has sent a new block. *).

5.3.2 Read Status (Block 9250)

This block is used to request status data from the module by the processor.

Block Format from Processor (4x Register Data) sent to module

Word Offset in Block	Data Field(s)	Description
0	Sequence Counter	This field contains a new value each time the user wishes to request a new command block.
1	Block ID	This field contains the block identification code of 9250 for the block.
2 to 63	Spare	Not used.

Block Format from Module (3x Register Data) from the module to Processor

Word Offset in Block	Data Field	Description
0	Scan Count	This status value contains a counter incremented on each scan of the module's main loop.
1 to 2	Product Name	This two-word data area contains the text values representing the product name. These words contain the text 'I4MQ' for the PTQ platform.
3 to 4	Revision	This two-word data area contains the text values for the revision number.
5 to 6	Op Sys #	This two-word data area contains the text values for the operating system number.
7 to 8	Run Number	This two-word data area contains the text values for the run number.
9	Read Blk Cnt	This word contains the total number of block read operations successfully executed.
10	Write Blk Cnt	This word contains the total number of block write operations successfully executed.
11	Reserve	This word is reserved for future use.
12	Error Blk Cnt	This word contains the total number of block transfer errors.
13	Event Msg Cnt	This word contains the number of event messages waiting to send to the processor.
14	Event Msg Overflow	This word contains a value of 0 if the event message buffer has not overflowed. If the event buffer overflows, this word will be set to a value of 1.
15	Session Count	This word contains the number of sessions configured in the module.
16	Current Cmd	This word contains the index of the current command being executed in the command list.
17	Cmd Busy Flag	This word is set to zero if no command is currently being executed and waiting on a response. If the word is set to 1, a command is currently executing.

18	Cmd Count	This word contains the count of the number of commands configured for the module.
19	Cmd Delay	This word contains the command delay counter preset. There is a fixed delay between each command to permit the module to perform other operations.
20	Cmd Queue	This word is set to zero if the command executing is from the command list. If the executing command is from the command queue, the word will be set to 1.
21	Cmd Queue Count	This word contains the number of active commands in the command queue for the module. Up to 100 commands can be buffered in this queue. These commands are transferred from the processor to the module using special command blocks.
22 to 23	Online Status	This double word value contains a bit for each of the 32 potential sessions in the module. If the bit is set for a session in the double word, the station is online. If the bit is clear, the station is offline. Use this value to determine if commands sent from the processor will have a chance of succeeding.
24	SNTP Valid	NTP time is valid (0=No, 1=Yes)
25	NTP Request	Number of requests to NTP server
26	NTP Response	Number of responses from NTP server
27	SNTP Computation	Number of times SNTP time computed
28	SNTP Set	Number of times SNTP time set
29	NTP Timeout	Number of NTP response timeouts

5.3.3 Event Messages (9958)

Block 9958 is reserved to send event messages from the processor to the module. Each block can send up to 10 events to the module. If a value of 9958 is placed in the control register, event messages are sent from the processor to the module. Refer to Events (page 156) for more information about timestamped events.

The module supports a buffer queue of 99 events per data type. When the queue is full, the module will delete the older event in the queue if a new event is received.

This block should only be used to pass events with a predefined timestamp (the module will also send timestamped events when database values change). While using block 9958, disable the events for those specific points (refer to Group Interrogation table) to avoid multiple event generation (caused by point value update through the database).

The following table shows the block format for write.

Block Format for Write

Word Offset in Block	Data Field(s)	Description
1	Block ID	This field contains the value of 9958 identifying the block type to the module.
2	Event Count	Number of events present in the block. Valid Values: 1 to 4.
3	Event #1	Event data to add to event message queue.
18	Event #2	Event data to add to event message queue.
33	Event #3	Event data to add to event message queue.
48	Event #4	Event data to add to event message queue.

The structure of each event record in the block is shown in the following table.

Word Offset in Event Record	Data Field(s)	Description
0	DB Index	This is the index for the point in the module's database. This corresponds to the order of point definition for the module data types. This is not the point address for the event. For example, to refer to the third configured point, enter a value of 2 (first event is 0).
1	ASDU	This is the ASDU data type for the event message. Valid entries for this field are as follows: 1=single-point 3=double-point 5=step 9=normalized 11=scaled 15=integrated total
2	Qualifier	This is the qualifier code for the event message. This parameter is only used for ASDU types 5, 9, 11 and 15. Refer to the IEC Protocol Specification for a full listing of valid qualifier codes for one of these ASDU types.
3	Year	This field contains the four-digit year for this event.
4	Month	This field contains the month value for the event. Valid Values: 1 to 12
5	Day	This field contains the day value for the event. Valid Values: 1 to 31
6	Hour	This field contains the hour value for the event. Valid Values: 0 to 23
7	Minute	This field contains the minute value for the event. Valid Values: 0 to 59
8	Seconds and Milliseconds	This field contains the seconds and milliseconds value for the event. Valid Values: 0 to 59,999 (59 seconds and 999 milliseconds).
9 to 14	Data	These words contain the data for the event. For single- and double-point, step, and measured value events, the first word is used. For integrated total events, the first two words are used.

Block Format for Read

Word Offset in Block	Data Field(s)	Description
1	Block ID	This word will contain the value of 9958

5.3.4 Read Module's Time to Processor (9970)

If a value of 9970 is placed in the control register, the processor will read the module's current time.

Block Format for Write

The following table shows the block format for write:

Word Offset in Block	Data Field(s)	Description
1	Block ID	This field contains the value of 9970 identifying the block type to the module.

Block Format for Read

The module responds to a valid 9970 request with a block containing the requested date and time. The block format is shown in the following table.

Word Offset in Block	Data Field(s)	Description
1	Block ID	This word will contain the value of 9970.
2	Year	This field contains the four-digit year for the new time value.
3	Month	This field contains the month value for the new time. Valid Values: 1 to 12.
4	Day	This field contains the day value for the new time. Valid Values: 1 to 31.
5	Hour	This field contains the hour value for the new time. Valid Values: 0 to 23
6	Minute	This field contains the minute value for the new time. Valid Values: 0 to 59.
7	Seconds	This field contains the second value for the new time. Valid Values: 0 to 59.
8	Milliseconds	This field contains the millisecond value for the new time. Valid Values: 0 to 999.

5.3.5 Block 9971: Set Module Time

If a value of 9971 is placed in the control register, Module time is set using the processor's time. The following table shows the block format for write.

Block 9971: Block Format for Write

Word Offset in Block	Data Field(s)	Description
0	Write Block ID	This word will contain the value of 9971.
1	Year	This field contains the four-digit year for the new time value.
2	Month	This field contains the month value for the new time. Valid Values: 1 to 12.
3	Day	This field contains the day value for the new time. Valid Values: 1 to 31.
4	Hour	This field contains the hour value for the new time. Valid Values: 0 to 23
5	Minute	This field contains the minute value for the new time. Valid Values: 0 to 59.
6	Seconds	This field contains the second value for the new time. Valid Values: 0 to 59.
7	Milliseconds	This field contains the millisecond value for the new time. Valid Values: 0 to 999.

Block Format for Read

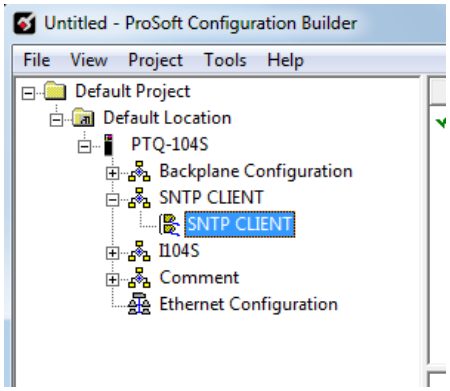
Word Offset in Block	Data Field(s)	Description
1	Block ID	This word will contain the value of 9971

5.3.6 Block 9998 or 9999: Reboot Module

If the processor places a value of 9998 in this register, the module will perform a warm-boot operation. If the processor places a value of 9999 in this register, the module will perform a cold-boot operation. In this application module, both of these operations perform the same function. They exit the program and then restart the program. Many of the program parameters set in the user configuration must be set at program initialization and cannot be set while the program is running. Therefore, both functions operate the same way.

5.4 [SNTP CLIENT]

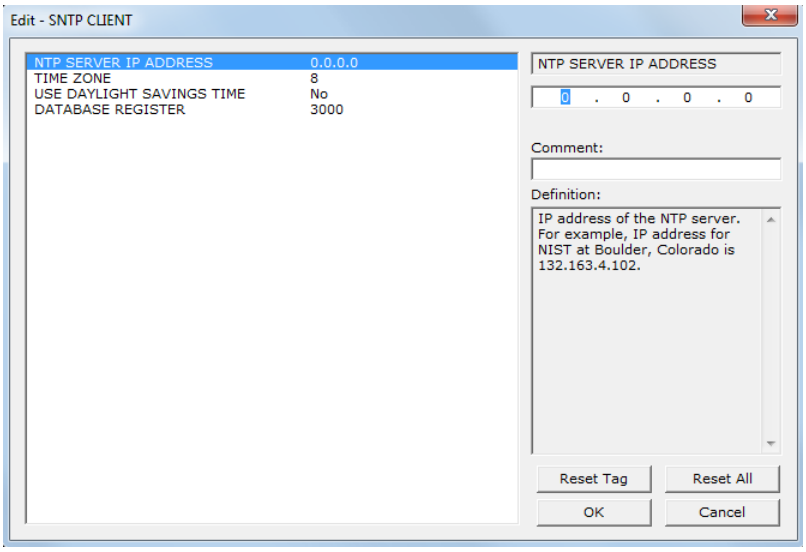
The [SNTP CLIENT] section is used to specify the parameters for the Simple Network Time Protocol (SNTP) Client provided with the protocol driver.



This client is read in order to keep the driver's internal clock set correctly. This version of the driver supports SNTP Revision 3 and stratum between 1 and 14.

SNTP is used for time synchronization of produced and consumed commands. When an exchange occurs the driver compares time stamps from the previous exchange. When the new exchange time is less than the previous exchange, the exchange is ignored. This can occur when the Ethernet packets are routed and delayed. Time synchronization provides for data integrity.

```
# This section used to define the parameters required for the Simple Network
Time
# Protocol (SNTP) client.
[SNTP CLIENT]
#NTP SERVER IP ADDRESS    : 132.163.4.102  #IP address for NIST, Boulder,
Colorado
NTP SERVER IP ADDRESS     : 0.0.0.0        #IP Address for NTP Server
TIME ZONE                 : 8              #Number of hours from GMT (-11 to
+11)
USE DAYLIGHT SAVINGS TIME : No             #Yes or No
DATABASE REGISTER         : 3000           #database word location to store
time
                                    #(-1=ignore). This register value should
                                    #be an even number.
```



The SNTP driver will compute a new clock value every 5 minutes using the average value of 10 samples each collected over an approximate 6-second period. This new value will be used to adjust the clock maintained by the SNTP driver and used by the application. If a valid database register is specified, the driver will place the time value into the module's database. The first two registers will contain the number of seconds and the next two registers will contain the number of microseconds since January 1, 1970.

A list of some of the common NTP servers can be obtained at http://www.ntp.org/

or, http://support.ntp.org/bin/view/Servers/WebHome

Other server lists can be found by searching the World Wide Web for "NTP Servers".

5.4.1 NTP Server IP Address

Enter in dotted notation

This parameter sets the IP address of the NTP server to utilize for time acquisition. Select an NTP server with the greatest accuracy that can be accessed all the time from your network. Setting this IP address to 0.0.0.0 disables SNTP server requests.

5.4.2 Time Zone

-11 to 11

This parameter specifies the time zone offset to be used from the UTC time zone. A value of zero uses UTC time. If the value entered is positive, the time zone is west of the UTC time zone (that is, Eastern Standard Time is 5). If the value entered is negative, the time zone is east of the UTC time zone (that is, Continental Europe is -1).

5.4.3 Use Daylight Savings Time

Yes or No

This parameter specifies if daylight savings time will be used in the time computation.

5.4.4 Database Register

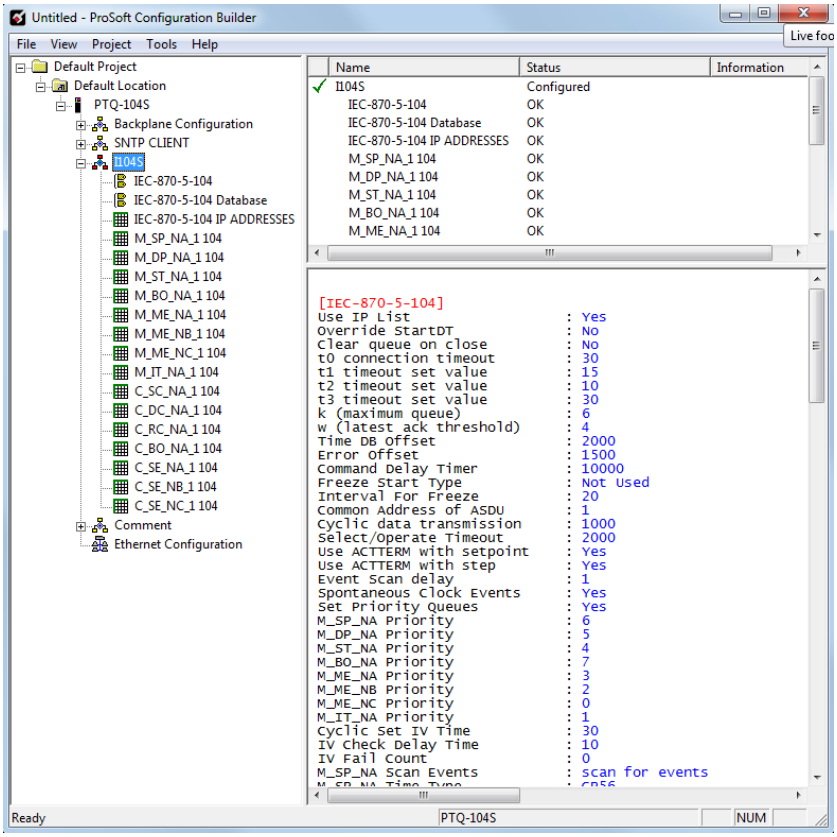
-1 or 0 to 3992 as an even value

This parameter specifies if the NTP time computed by the driver is to be placed into the module's database. If a value of -1 is specified, the time will not be placed into the database. If the value is between 0 and 3992, the time will be placed in the database. The first 4 bytes will represent the seconds since 1/1/1970, and the second 4 bytes will represent the number of microseconds. An even value should be used for the register value in order for the data to be stored correctly.

5.5 [I104S] Section

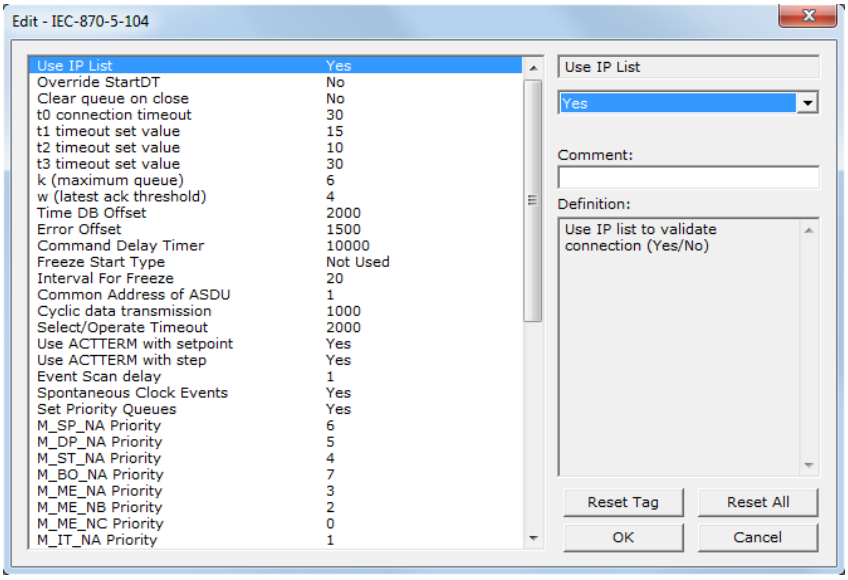
The I104S section allows the user to setup the following features:

- General Server parameters
- General Database parameters
- IP Access List addresses
- IEC Database points for each data type



5.6 [I104S] > [IEC-870-5-104]

This section provides information required to configure a server application with the module. An item of concern is the maximum size of the total database. Although it is possible to configure a database of considerable size, the maximum Class 0 request may not exceed 2048 bytes in size.



5.6.1 Use IP List

No (0) or **Yes (1)**

This parameter specifies if the IP address of the host connected to the system will be validated. If the parameter is set to **No (0)**, any host may connect to the unit. If the parameter is set to **Yes (1)**, only hosts in the IP list will be permitted to connect to the unit.

5.6.2 Override StartDT

No (0) or **Yes (1)**

This parameter is used when testing the module with a simulator or with a Client unit that does not meet the IEC 60870-5-104 specification. After the Client connects to the module, it will send a STARTDT.ACT U-format message to the module to permit the module to start sending data. If the Client does not support this requirement, set the parameter to **Yes (1)**. Set the parameter to **No (0)** if the Client sends the STARTDT.ACT message.

Note: This parameter must be set to **No (0)** if you wish to use the redundant Clients feature.

5.6.3 Clear Queue on Close

NO (0) or **YES (1)**

Use this command to define whether the module will store the unacknowledged buffers in the unit after the connection is closed. If the specification is to be followed, set this parameter to **NO (0)** and the packets will be resent after a connection is made. If you want to flush the packets after the connection is closed, set this parameter to **YES (1)** (this is not according to the IEC 60870-5-104 specification).

5.6.4 t0 Connection Timeout

1 to **1000**, default value = **30**

This is a timeout value, in seconds, to determine if a connection has been lost with the remote Client. If no traffic from the remote client is received for the period of time specified by this parameter, the currently open server IP socket connection will be closed. The connection can be re-established by the remote Client by opening a new connection.

5.6.5 t1 Timeout Set Value

1 to **255**

This is the timeout of send or test ASDUs in seconds. After a packet is sent from the unit, the Client must acknowledge the packet within this time interval or else the unit will close the connection.

5.6.6 t2 Timeout Set Value

1 to **255**

This is a timeout of when to send an S-format message to the host to acknowledge outstanding messages received. This parameter is in units of seconds and must be less than the value set for t1.

5.6.7 t3 Timeout Set Value

1 to **255**

This is the timeout to wait on an idle line before the unit will send a TestFr.Act message. This value is in units of seconds.

5.6.8 k (maximum queue)

1 to **20**

This parameter specifies the number of unacknowledged messages the unit will buffer. This parameter must match that in the host. If the set number of buffers are filled in the unit, no other messages will be sent until the host unit acknowledges some or all the messages.

5.6.9 w (latest ack threshold)

1 to **20**

This parameter must match that of the host unit and specifies the number of messages the module will receive before sending an S-format sequence acknowledge message when no I-format data is ready to send. It is recommended to set this value to 2/3 the value of k.

5.6.10 Time DB Offset

-1 or **0** to **3994**

This parameter sets the location in the database where the module's current date and time will be copied to.

Note: The following tables lists the 12-byte data area placed in the database if the Time DB Offset parameter is set to a value other than -1:

Byte	Length	Range	Description
0 to 1	2	0 to 59,999	Seconds and milliseconds
2	1	0 to 59	Minutes
3	1	0 to 23	Hour
4	1		Reserved
5	1	1 to 31	Day of the Month
6	1	1 to 12	Month
7 to 8	2	0 to 65,535	Year (four digit format, for example 2005)
9	1		Reserved
10	1	0 or 1	Invalid Flag (0 = Valid, 1 = Invalid
11	1		Reserved

5.6.11 Error Offset

0 to 3980

-1 to disable

This parameter sets the location in the module's internal database when the status and error information for the server will be stored. Enter a valid database address of 0 to 3980 to store the information, or enter -1 to disable.

5.6.12 Command Delay Timer

1000 to **60000** milliseconds

This value is used for time-tag validity verification for the ASDUs listed below. The received commands will only be accepted if [timestamp + Command Delay Timer] is greater than the module's own time.

- 58: Single command with time tag CP56Time 2a
- 59: Double command with time tag CP56Time 2a
- 60: Regulating step command with time tag CP56Time 2a
- 61: SetPoint command, normalized value with time tag CP56Time 2a
- 62: SetPoint command, scaled value with time tag CP56Time 2a
- 63: SetPoint command, short floating point with time tag CP56Time 2a
- 107: Test command with time tag CP56Time2a

If the value is less than **1000** milliseconds, the module will default to **5000** milliseconds

5.6.13 Freeze Start Type

D = **DAY, H** = **HOUR, M** = **MINUTE, N** = **NOT USED**

The Freeze Start Type parameter defines when the module starts sending the M_IT messages.

5.6.14 Interval for Freeze

0 to **65535** seconds

Freeze Start Type and *Interval for Freeze* are used if Mode A operation is to be used for the counter freeze operation. If they are not used, the module will operate in Mode D.

5.6.15 Common Address of ASDU

0 to **65535**

This parameter specifies the common address of the ASDU (section address) for access to data in the module. There is only one value entered for access to all data in the module.

5.6.16 Cyclic Data Transmission

0 to **4,294,967,296** (2 raised to the power of 32)

This parameter defines the number of milliseconds between cyclic updates. The range of values for this parameter permits update times of 1 millisecond to 5 minutes. If the parameter is set to 0, cyclic data reporting will be disabled.

5.6.17 Select/Operate Timeout

0 to **4,294,967,296** (2 raised to the power of 32) milliseconds

This parameter sets the number of milliseconds to wait for a valid execute command after receiving a select command. The range of values allow for update times between 1 millisecond and 49.7 days. Set this parameter to 0 to disable this feature.

5.6.18 Use ACTTERM with Setpoint

Y - YES or **N - NO**

This parameter determines if an ACTTERM will be sent. If the parameter is set to Yes, setpoint commands will issue an ACTTERM when the command is complete. If the parameter is set to No, ACTCON is the last response to a setpoint command.

5.6.19 Use ACTTERM with step

Y - YES or **N - NO**

This parameter determines if an ACTTERM will be sent. If the parameter is set to Yes, then step commands will issue an ACTTERM when the command is complete. If the parameter is set to No, ACTCON is the last response to a step command.

5.6.20 Event Scan Delay

1 to **65535** milliseconds, or **0** to disable

If set to 0, the feature will be disabled and the module will not generate any events. If set from 1 to 65535, the parameter represents the number of milliseconds between event scanning. This parameter defines how often the program will scan for new events in the databases.

5.6.21 Spontaneous Clock Events

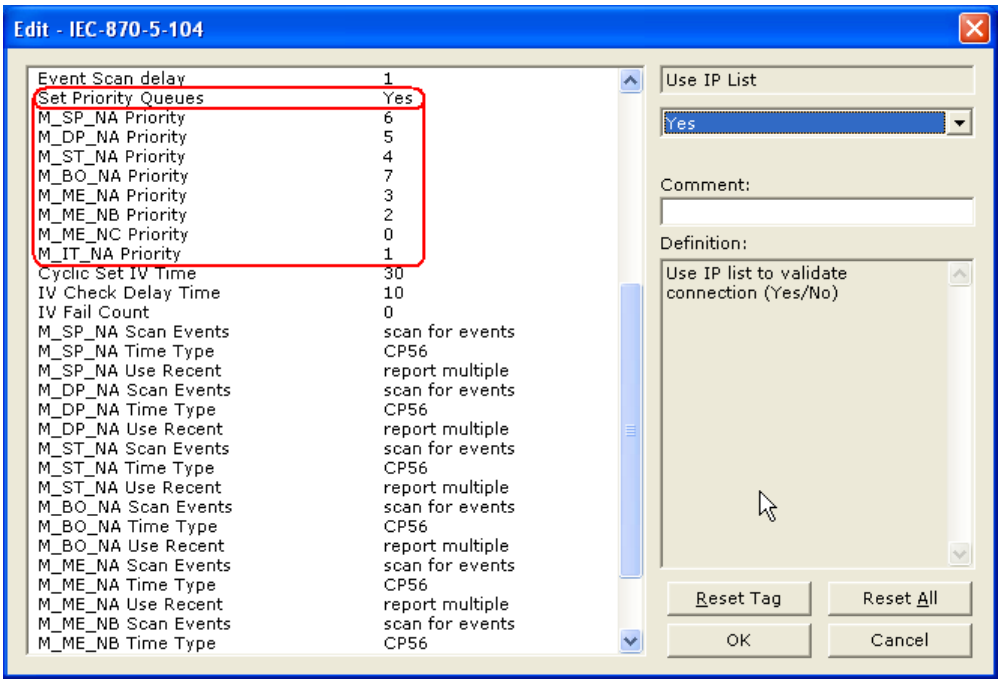
Y - YES or **N - NO**

If the parameter is set to Yes, then the module will generate spontaneous (COT=3) clock sync event messages. If the parameter is set to No, then the module will not send the clock sync event messages

5.6.22 Set Priority Queues

YES or **NO**

This section defines priority queues for the module. You can assign priorities to data types that can return events so that events of M_IT_NA data types will be returned before other data types. This may cause events to be lost as the event buffers for low priority queues may overflow. If this feature is utilized, each data type must be assigned a unique index from 0 to 7. The lower the index, the higher the priority (0=highest priority).



Each of the ASDUs affected by this feature must be assigned a unique priority index from 0 to 7. Events of the ASDU with a priority of 0 will always be reported before any others when they are present.

For more information, refer to Event Priority (page 167).

5.6.23 Cyclic Set IV Time

1 to **65535** seconds, or **0** to disable Invalid Bit Monitoring

The parameter should be set to a value significantly greater than the value of the *IV Check Delay Time* parameter, multiplied by the value of the *IV Fail Count* parameter. Example:

If, *IV Check Delay* = **3**
And, *IV Fail Count* = **5**;
Then set *Cyclic Set IV Time* significantly greater than 15; preferably some multiple of this value, like **30**, **45**, **60,** or more.

The *Cyclic Set IV Time* parameter is one of three parameters needed to establish a fail-safe data validation system for this protocol implementation. This parameter is used with the *IV Check Delay Time* and the *IV Fail Count* parameters to create a way to alert the remote IEC-104 Client to situations where data being sent by the module might be invalid.

Use the *Cyclic Set IV Time* parameter to set how frequently the Invalid (IV) Bits assigned to IEC database monitor points (M_xx_xx points) will be set ON. This means that all the IEC database monitor points configured with IV Bit addresses greater than zero (0) will have their IV Bit addresses in the module database set to a value of one (1) in a timed, periodic cycle, based on the number of seconds entered in *Cyclic Set IV Time*.

The IV Bits are checked at the interval specified by the *IV Check Delay Time* parameter. At the end of each check delay interval, if an IV Bit for any IEC database point is ON, an IV Bit Check Fail accumulator for that point will be incremented. If the value in any IV Bit Check Fail accumulator becomes equal to the value of the *IV Fail Count* parameter, the module will consider the data value of that point invalid and will report the invalid status of that point to the Client along with the data value.

If the IV Bit is cleared (reset, set OFF, set to zero) before the IV Bit Check Fail accumulator value becomes equal to the value of the *IV Fail Count* parameter, the IV Bit Check Fail accumulator will be reset to 0. The module will consider the data value of that point valid and will report the data value to the Client as valid.

To properly use data validity checking with a ProTalk rack-mounted module for Modicon Quantum processor systems, the logic application running in the processor that communicates to the module through a backplane input/output driver must be programmed to provide data validity information, as well as the actual data to be transferred. This logic application must write bit values of zero (0) or one (1) to the proper module database addresses along with the data for each IEC database point.

This means the other protocol application or the logic application will be responsible for clearing the IV Bits for data points which contain valid data at a time interval less than the amount of time required to accumulate the required number of IV Bit Check failures. In cases where ProTalk modules are used, backplane communication happens so quickly that cyclic setting of the IV Bits will be quickly undone by data transfers from processor memory. However, if the process logic or other protocol application does not clear the IV Bits set by firmware, all IEC database monitor point values will be reported to the Client as invalid.

NOTE: If one or more of the *Cyclic Set IV Time*, the *IV Check Delay Time*, or the *IV Fail Count* parameters are set to zero, the data validity checking feature will be disabled for all points in the IEC database and the data reported to the remote Client will always be reported as valid data. If you wish to have actual data validity information relayed to the Client, be sure the *Cyclic Set IV Time*, the *IV Check Delay Time*, and the *IV Fail Count* parameters are set to non-zero values and be sure the application has been programmed to properly manipulate the IV Bit addresses in the module database. Once data validity checking has been enabled, you can still turn off validity checking on a point-by-point basis by setting the IV Bit Address to zero (0) for any point you wish to exclude from data validity checking and reporting.

For more information on the data validity features of this protocol implementation, refer to Invalid Bit Monitoring (page 163).

5.6.24 IV Check Delay Time

1 to **65535** seconds, or **0** to disable Invalid Bit Monitoring

This parameter sets the number of seconds between IV Bit value checks. Every IEC database monitor point which has an IV Bit address set greater than 0 will have that bit address checked at the interval specified by this parameter.

Setting this parameter to zero (0) will disable data validity checking.

5.6.25 IV Fail Count

1 to **65535** seconds, or **0** to disable Invalid Bit Monitoring

This parameter sets the number of successive IV Bit check failures which must occur before the data from a IEC database monitor point will be reported to the remote Client as invalid data.

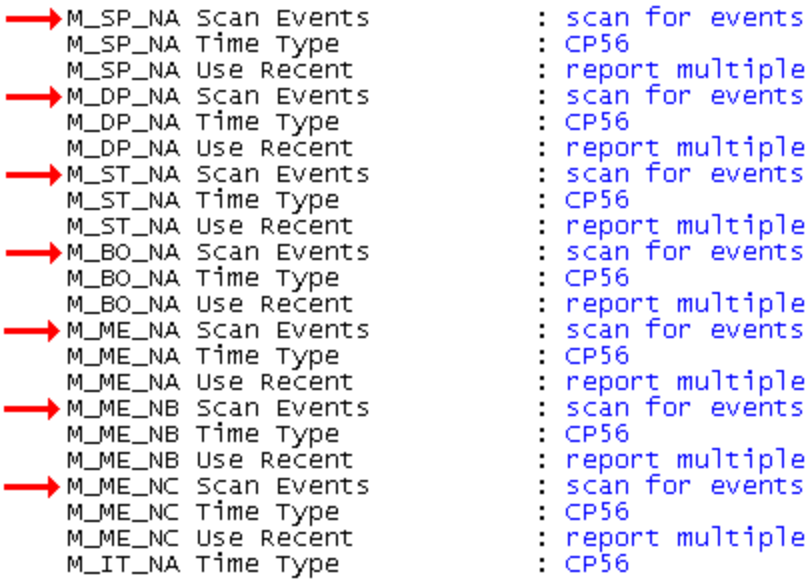
An IV Bit Check Failure occurs when the IV Bit value in the module database is set ON {when it contains a value of one (1)} at the time an IV Bit Check is performed. IV Bit Check failures are counted and held in separate IV Bit Check Failure accumulators for each IEC monitor point configured for validity checking. If the value in any point's failure accumulator becomes equal to the value set in this **IV FAIL COUNT** parameter, the module will consider data from this point as invalid and report to the Client this invalid status, along with the point's data value.

Setting this parameter to zero (0) will disable data validity checking.

5.6.26 Scan Events

0 = **NO SCANNING** or **1** = **SCAN FOR EVENTS**

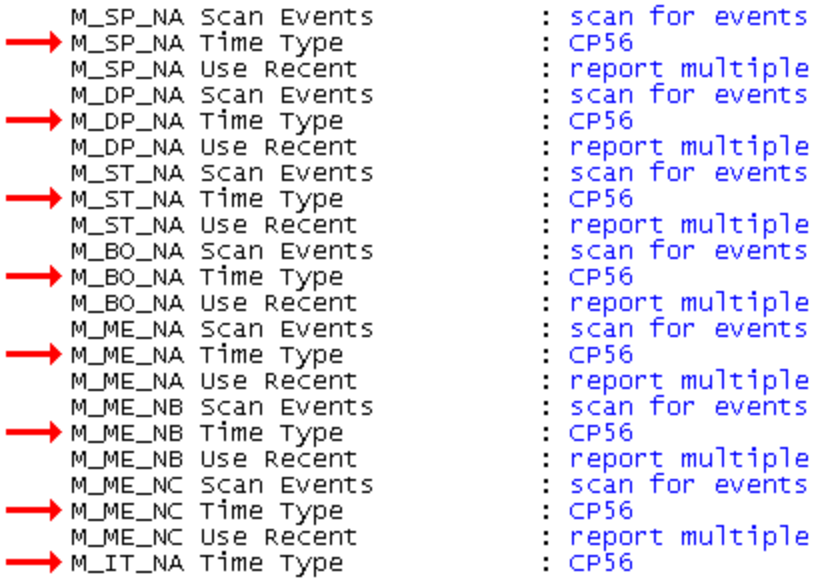
Separate parameters exist for multiple point types. Defines whether events of this point type will be generated by the module. If set to zero (**0**) for **NO SCANNING**, then events will not be generated. If set to **SCAN FOR EVENTS**, events will be scanned and generated on data change.



5.6.27 Time Type

0 = **NONE**, **1** = **CP24**, or **2** = **CP56**

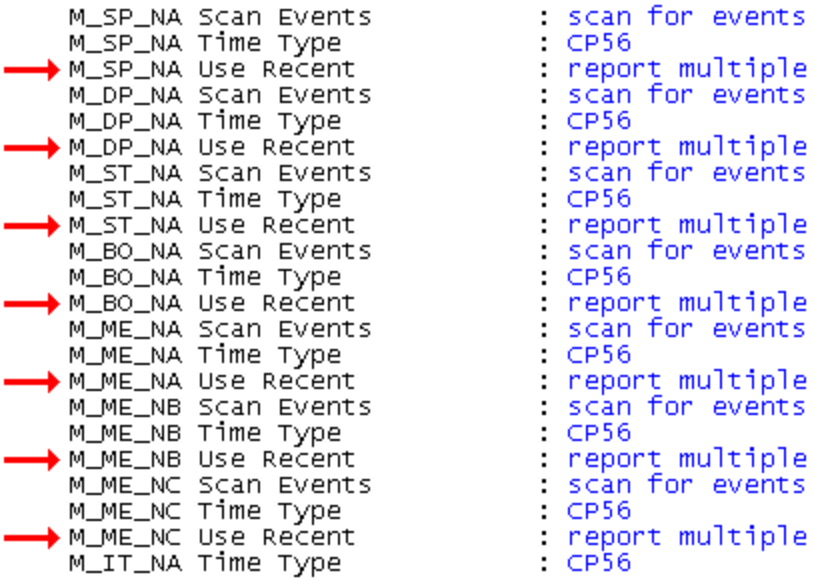
Separate parameters exist for multiple point types. This parameter defines the time format used with data events.



5.6.28 Use Recent

0 = **REPORT MULTIPLE**, **1** = **REPORT SINGLE**

This parameter determines how the events will be stored at the event queue and reported to the remote Client. The 104S protocol supports event queues for each data type that can store multiple, time-stamped events as they happen and report those events to a remote Client. Some applications may require all events that transpire be added to the queue and be reported. Other applications may require only the most recent event of each IEC database point to be reported.

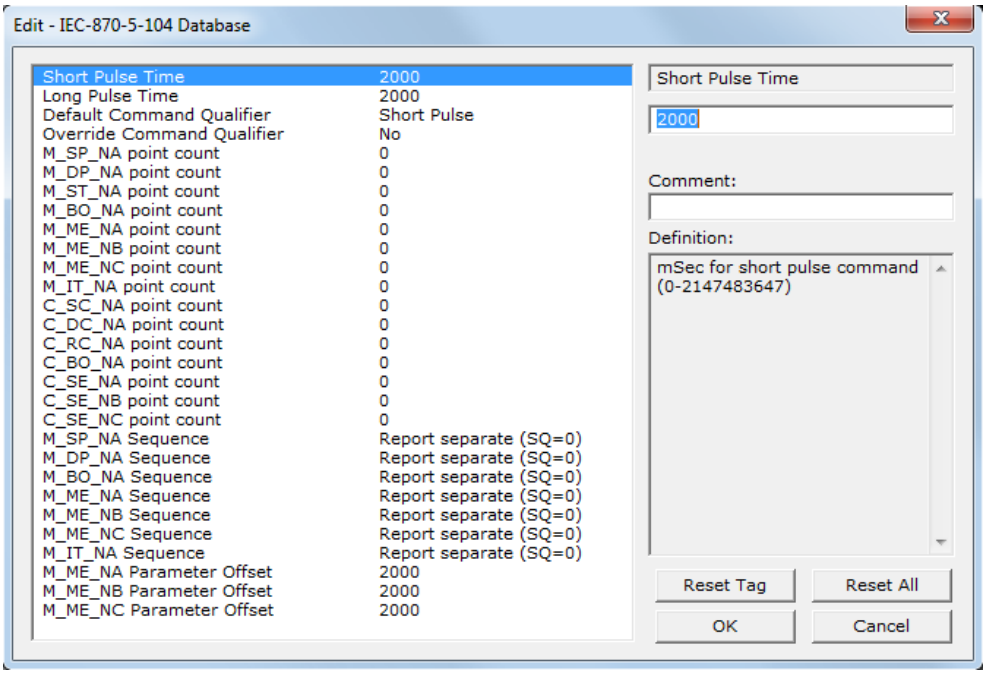


If this parameter is set to **0** - **REPORT MULTIPLE** (the default), events for each configured IEC monitor-type data point will be stored in a buffer. Each monitor point data type has its own unique buffer that can hold to 99 events. When the events are reported to the remote Client, the buffer will be cleared and will begin to accumulate new events as they happen, storing them until the next time they can be reported. If the number of events that occur between reports exceed 99 for any given data type, then the oldest event entry in the queue will be deleted to make room for the most recent event. The advantage to this setting is that, with proper configuration, all events will be captured and reported to the Client. The disadvantage is that the event buffers can contain high volumes of data that will have to be transferred to the Client.

If this parameter is set to **1** - **REPORT SINGLE**, only the most recent event for each IEC database monitor point will be held in the buffer and reported to the Client. This means that any events other than the most recent one for each point will not be available. The advantage to this setting is that it limits the volume of event data that may need to be transmitted to the Client to the total number of points of all types in the IEC database. The disadvantage is that events for some points may be lost and not reported to the Client.

5.7 [I104S] > [IEC-870-5-104 Database]

This section describes parameters in the [I104S]>IEC-870-5-104 Database section.



5.7.1 Short Pulse Time

0 to **2,147,483,647** (2 raised to the power of 31) milliseconds

This parameter defines the number of milliseconds to keep an IEC Command Point set ON, HIGH, equal to one (1), before returning the point to OFF, LOW, equal to zero (0), whenever a pulse point command is sent by the Client and the default value of the *Default Command Qualifier* parameter is set to **SHORT**.

5.7.2 Long Pulse Time

0 to **2,147,483,647** (2 raised to the power of 31) milliseconds

This parameter defines the number of milliseconds to keep an IEC Command Point set ON, HIGH, equal to one (1), before returning the point to OFF, LOW, equal to zero (0), whenever a pulse point command is sent by the Client and the default value of the *Default Command Qualifier* parameter is set to **LONG**.

5.7.3 Default Command Qualifier

SHORT PULSE (1), **LONG PULSE** (2), or **PERSISTENT OUTPUT** (3)

The protocol specification defines values for a *Qualifier Flag*, which is part of every command message, and associates those values to specific command (control) points. The value zero (0) is specified as *Undefined*. The *Default Command Qualifier* parameter sets the default output type to use for incoming Command Point executions when the *Qualifier Flag* value is *Undefined*.

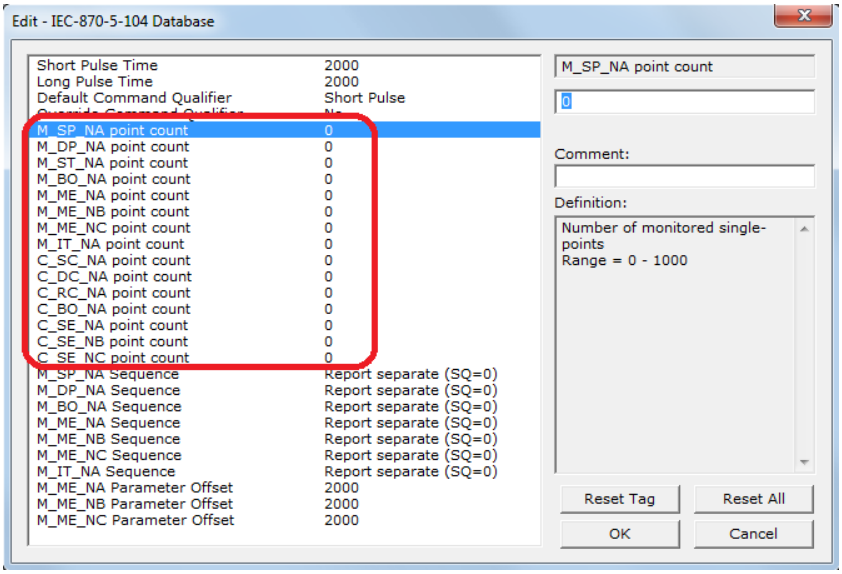
5.7.4 Override Command Qualifier

NO, **SHORT PULSE**, **LONG PULSE**, or **PERSISTENT OUTPUT**

This parameter allows you to set a specific output type for all incoming command messages, regardless of the *Qualifier Flag* value contained in the message. Set this parameter to **NO** to use the output type specified by the *Qualifier Flag* received in the command message. If this parameter is set to any of the other options, the value received in the *Qualifier Flag* is ignored and the selected output type will be used for all command executions.

5.7.5 Point Count

Point Count configuration ranges in the following configuration items are based on the assumption that you will be using only one of the available data types for your application. The number of point counts you configure will have an effect on module performance, in particular the accuracy of the module's internal clock.



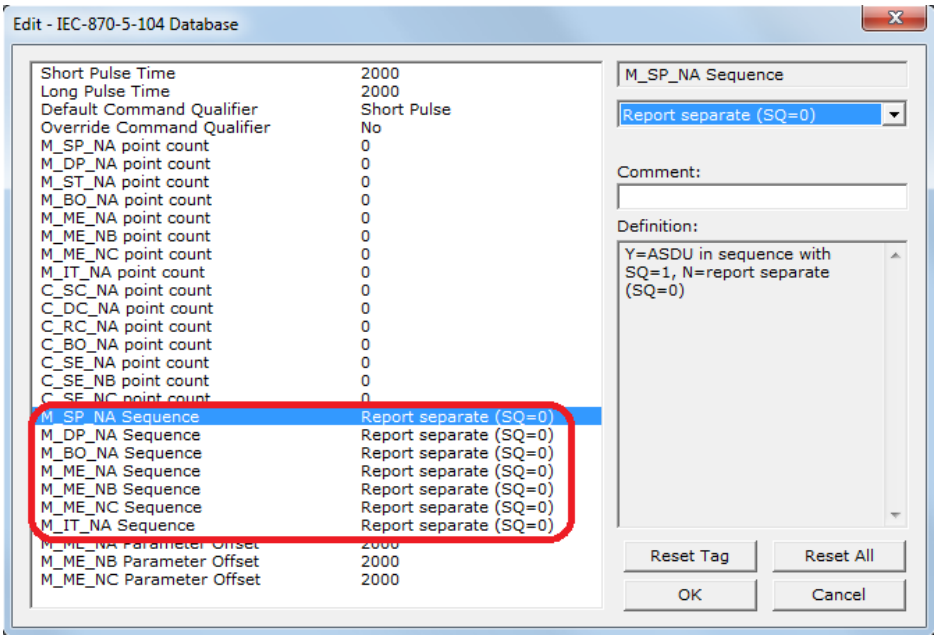
These parameters specify the number of point values that can be assigned for each database.

Type ID	Point Database Description	Range
M_SP_NA	Monitored Single-Point	0 to 1000
M_DP_NA	Monitored Double-Point	0 to 1000
M_ST_NA	Monitored Step Position Point	0 to 1000
M_BO_NA	Monitored 32-bit Bitstring Data Point	0 to 1000
M_ME_NA	Monitored Normalized Measured Point	0 to 1000
M_ME_NB	Monitored Scaled Measured Point	0 to 1000
M_ME_NC	Monitored Short Floating-Point Measured Point	0 to 50
M_IT_NA	Monitored Integrated Total	0 to 99
C_SC_NA	Command Single-Point	0 to 1000
C_DC_NA	Command Double-Point	0 to 1000
C_RC_NA	Command Step Point	0 to 1000
C_BO_NA	Command 32-bit Bitstring Point	0 to 1000
C_SE_NA	Command Normalized Set Point	0 to 1000
C_SE_NB	Command Scaled Set Point	0 to 1000
C_SE_NC	Command Short Floating-Point Set Point	0 to 50

5.7.6 Sequence Flag

YES or **NO**

In order to save bandwidth, you can configure the module to use the Sequence Flag feature using the following parameters:

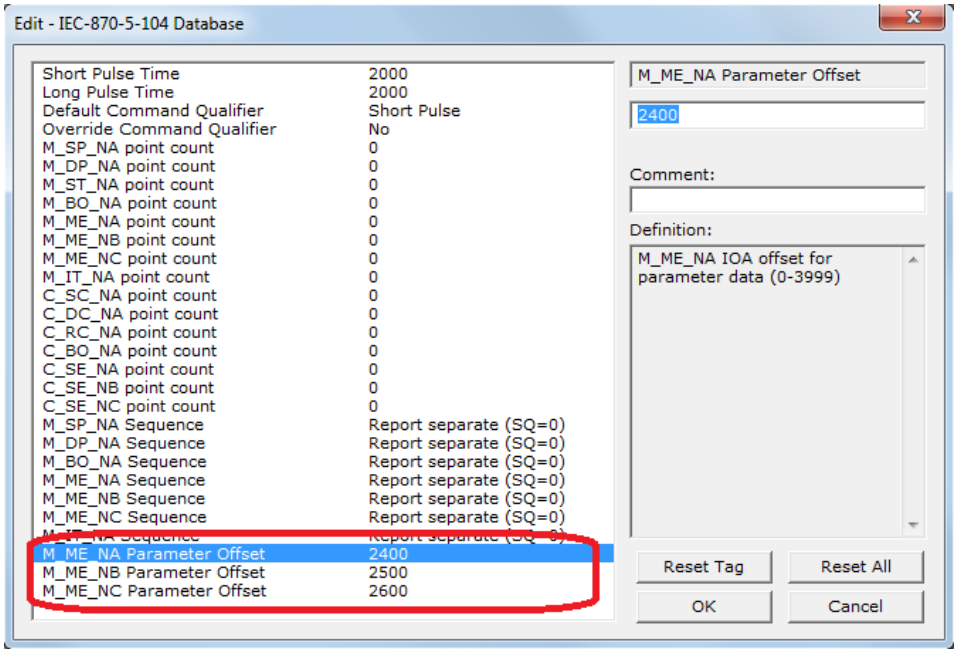


If this parameter is set to **No**, this feature is not selected, the module will send the object address and value for every monitored point sent to the Client, and the Sequence Flag (SQ) will be set to zero (0).

If this parameter is set to **YES**, this feature is selected, the module will send the object point address and value for only the first point of a sequence of points, send only the data value without point address for any remaining points in the sequence, and the Sequence Flag (SQ) will be set to one (1). When SQ = 1, the Client assumes that all points after the first point use Information Object Addresses in a contiguous order (using the first point as the reference starting address).

Note: Refer to the Client device specification to verify if this feature is supported before you consider using it.

5.7.7 Parameter Offset



This parameter specifies the Information Object Address (IOA) Offset to normalized points parameter data. The value entered is added to the IOA for the associated point to compute the parameter IOA address.

The Client may send a "Parameter of Measured Normalized" or "Parameter of Measured Scaled" command using the parameter IOA in order to change the threshold deadband values for specific points.

For example, if the PTQ-104S configuration sets two M_ME_NA points with IOA of 600 and 601 and a M_ME_NA Parameter Offset value of 2000, the threshold deadband parameters for those points would be written to IOA addresses 2600 and 2601.

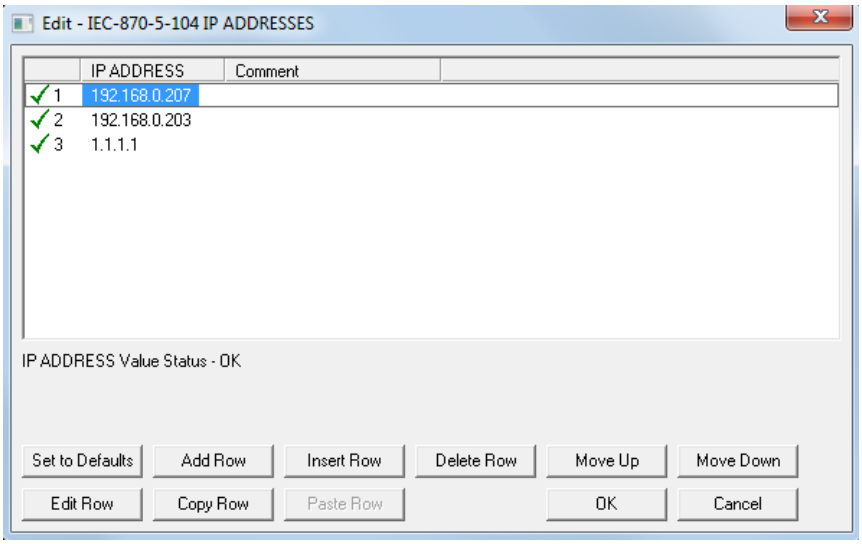
Note: The Low Limit and High Limit threshold values are always calculated based on the threshold deadband value as described in the following table.

Point	Value
Threshold	Determined by the deadband set in the configuration file or altered by the write command.
Low Limit	Last reported event value - threshold.
High Limit	Last reported event value + threshold.

5.8 [I104S] > [IEC-870-5-104 IP ADDRESSES]

This section enters the IP addresses for the hosts to connect to this unit. The unit will only accept connections from hosts listed here. This list may contain up to 10 entries between the START and END labels. The address must start in column 1, and must be entered in standard dot notation.

The following is an example of the [IEC-870-5-104 IP ADDRESSES] section:



5.9 [I104S] > [M_SP_NA_1 104]

This section defines the monitored single-point database. This information is sourced from the database and is transferred to the remote Client unit. Each point indicates one of two states, 1 = Bit On, 0 = Bit Off.

This section takes the following parameters:

- Point #
- DB Address
- Group(s)
- IV DB Bit

Each point is one bit and the DB address value corresponds to the bit offset in the module memory database. For additional information on how to set these parameters, see the Reference chapter of this manual.

5.10 [I104S] > [M_DP_NA_1 104]

This section defines the monitored dual-point database. This information is sourced from the database and is transferred to the remote Client unit. Each point in the database can have one of four possible states, 00 = intermediate, 01 = Point Off, 10 = Point On, and 11 = intermediate).

This section takes the following parameters:

- Point #
- DB Address
- Group(s)
- IV DB Bit

Each point is two bits and the DB address value corresponds to the bit offset in the module memory database. For additional information on how to set these parameters, see the Reference chapter of this manual.

5.11 [I104S] > [M_ST_NA_1 104]

This section defines the monitored step database. This information is sourced from the database and is transferred to the remote Client unit.

This section takes the following parameters:

- Point #
- DB Address
- Group(s)
- IV DB Bit

Each point is one 8-bit byte and the DB Address value corresponds to the byte offset in the module memory database. For additional information on how to set these parameters, see the Reference chapter of this manual.

5.12 [I104S] > [M_BO_NA_1 104]

This section defines the 32-bit Bitstring database for the server device emulated. This information is sourced from the database and is transferred to the remote client unit.

This section takes the following parameters:

- Point #
- DB Address
- Group (page 92)
- IV DB Bit

Each point is four, 8-bit bytes (two, 16-bit words) and the **DB ADDRESS** value corresponds to the double-word offset in the module memory database.

The **DB ADDRESS** should be located in a database area that is being constantly moved from the Quantum to the Master.

For additional information on how to set these parameters, see the Reference chapter of this manual.

5.13 [I104S] > [M_ME_NA_1 104]

This section defines the monitored measured value, normalized database. This information is sourced from the database and is transferred to the remote Client unit. To determine the IOA (Information Object Address) for each object, add the Point # (IOA) in the following section to the value of the *M_ME_NA Parameter Offset* as entered in the *[IEC-870-5-104 Database]* section.

This section takes the following parameters:

- Point #
- DB Address
- Group(s)
- Default Deadband
- IV DB Bit

Each point is one 16-bit word and the DB Address value corresponds to the word offset in the module memory database. For additional information on how to set these parameters, see the Reference chapter of this manual.

5.14 [I104S] > [M_ME_NB_1 104]

This section defines the monitored measured value, scaled database. This information is sourced from the database and is transferred to the remote Client unit. To determine the IOA (Information Object Address) for each object, add the Point # (IOA) to the *M_ME_NB Parameter Offset* parameter value as entered in the *[IEC-870-5-104 Database]* section.

This section takes the following parameters:

- Point #
- DB Address
- Group(s)
- Default Deadband
- IV DB Bit

Each point is one 16-bit word and the DB Address value corresponds to the word offset in the module memory database. For additional information on how to set these parameters, see the Reference chapter of this manual.

5.15 [I104S] > [M_ME_NC_1 104]

This section defines the monitored short-float point database for the slave device emulated. This information is sourced from the database and is transferred to the remote client unit. Each point occupies 4-byte positions in the database. To determine the IOA (Information Object Address) for each object, add the Point # in the following section to the value of the M_ME_NC Parameter Offset parameter set in the previous section.

This section takes the following parameters:

- Point #
- DB Address
- Groups
- Default Deadband
- IV DB Bit

Each point is one word and the DB Address value corresponds to the word offset in the database.

The Database Address value should be located in a database area that is constantly being moved from the Quantum processor to the PTQ-104S module. Therefore, this value should be configured in an area that is used on a Backplane Command Function 1.

Refer to the Group Codes (page 92) section for a listing of Group Codes.

5.16 [I104S] > [M_IT_NA_1 104]

This section defines the monitored integrated totals (counter) database. This information is sourced from the database and is transferred to the remote Client unit.

This section takes the following parameters:

- Point #
- DB Address
- Group(s)
- IV DB Bit

Each point is two 16-bit words and the DB Address value corresponds to the double-word offset in the module memory database. For additional information on how to set these parameters, see the Reference chapter of this manual.

5.17 [I104S] > [C_SC_NA_1 104]

This section defines the single point command database. This information is sourced from the remote Client and is transferred to the database. You can associate a command with a monitored single-point database value to coordinate the command/monitor operation. You must enter the correct Monitor Point # and Monitor DB Address values in the table. If the Require Select parameter is not set to zero, a select command must be received before an execute command will be processed.

This section takes the following parameters:

- Point #
- DB Address
- Monitor Point #
- Monitor DB Address
- Require Select

Each point is one bit and the DB Address value corresponds to the bit offset in the module memory database. For additional information on how to set these parameters, see the Reference chapter of this manual.

5.18 [I104S] > [C_DC_NA_1 104]

This section defines the double point command database. This information is sourced from the remote Client and is transferred to the database. You can associate a command with a monitored double point database value to coordinate the command/monitor operation. You must enter the correct Monitor Point # and Monitor DB Address values in the table. If the Require Select parameter is not set to zero, a select command must be received before an execute command will be processed.

This section takes the following parameters:

- Point #
- DB Address
- Monitor Point #
- Monitor DB Address
- Require Select

Each point is two bits and the DB Address value corresponds to the bit offset in the module memory database. For additional information on how to set these parameters, see the Reference chapter of this manual.

5.19 [I104S] > [C_RC_NA_1 104]

This section defines the step command database. This information is sourced from the remote Client and is transferred to the database. The control value can be associated with a monitored point as described in the previous example.

This section takes the following parameters:

- Point #
- DB Address
- Monitor Point #
- Monitor DB Address

Each point is one 8-bit byte and the DB Address value corresponds to the byte offset in the module memory database. For additional information on how to set these parameters, see the Reference chapter of this manual.

5.20 [I104S] > [C_BO_NA_1 104]

This section defines the 32-bit Bitstring command database for the server emulated. This information is sourced from the remote Master and is transferred to the database.

You can associate a command with a monitored double point database value to coordinate the command/monitor operation. You must enter the correct Monitor Point # and Monitor DB Addr values in the table. If the Require Select parameter is not set to zero, a select command must be received before an execute command will be processed.

This section takes the following parameters:

- Point #

- DB Address
- Monitor Point #
- Monitor DB Address

Each point is four, 8-bit bytes and the DB Address value corresponds to the double-word offset in the module memory database.

The DB Address should be located in a database area that is being constantly moved from the Master to the Quantum.

For additional information on how to set these parameters, see the Reference chapter of this manual.

5.21 [I104S] > [C_SE_NA_1 104]

This section defines the normalized setpoint database. This information is sourced from the remote Client and is transferred to the database. You can associate a command with a monitored normalized database value to coordinate the command/monitor operation. You must enter the correct Monitor Point # and Monitor DB Addr values in the table. If the Require Select parameter is not set to zero, a select command must be received before an execute command will be processed.

This section takes the following parameters:

- Point #
- DB Address
- Monitor Point #
- Monitor DB Addr
- Require Select

Each point is one 16-bit word and the DB Address value corresponds to the word offset in the module memory database. For additional information on how to set these parameters, see the Reference chapter of this manual.

5.22 [I104S] > [C_SE_NB_1 104]

This section defines the scaled setpoint database. This information is sourced from the remote Client and is transferred to the database. You can associate a command with a monitored scaled database value to coordinate the command/monitor operation. You must enter the correct Monitor Point # and Monitor DB Addr values in the table. If the *Require Select* parameter is set to one (**1**) or **YES**, a Select command must be received from the Client before an execute command from the Client will be processed.

This section takes the following parameters:

- Point #
- DB Address
- Monitor Point #
- Monitor DB Addr
- Require Select

Each point is one 16-bit word and the DB Address value corresponds to the word offset in the module memory database. For additional information on how to set these parameters, see the Reference chapter of this manual.

5.23 [I104S] > [C_SE_NC_1 104]

This section defines the short-float setpoint database. This information is sourced from the remote Client and is transferred to the database. If the *Require Select* parameter is set to one (**1**) or **YES**, a Select command must be received from the Client before an Execute command from the Client will be processed.

This section takes the following parameters:

- Point #
- DB Address
- Monitor Point #
- Monitor DB Addr
- Require Select

Each point is two 16-bit words and the DB Address value corresponds to the double-word offset in the module memory database. For additional information on how to set these parameters, see the Reference chapter of this manual.

5.24 Group Codes

One aspect of the point configuration database that leads to confusion is the group definition field, which assigns each point to one or more interrogation groups. Use of interrogation groups permits the controlling unit to interface with a specific set of data. Refer to the IEC 60870-5-104 standard for a full discussion of interrogation groups. A specific group, Periodic data group, reports data points on a set frequency. The frequency is set in the **Cyclic Data Transmission** parameter in the configuration file. Remember that a point can be assigned to more than one group.

Group Code	Description
0x00000001	Interrogated by general interrogation (station or global)
0x00000002	Interrogated by group 1 interrogation
0x00000004	Interrogated by group 2 interrogation
0x00000008	Interrogated by group 3 interrogation (See **Note 1**, below)
0x00000010	Interrogated by group 4 interrogation
0x00000020	Interrogated by group 5 interrogation
0x00000040	Interrogated by group 6 interrogation
0x00000080	Interrogated by group 7 interrogation
0x00000100	Interrogated by group 8 interrogation
0x00000200	Interrogated by group 9 interrogation
0x00000400	Interrogated by group 10 interrogation
0x00000800	Interrogated by group 11 interrogation
0x00001000	Interrogated by group 12 interrogation
0x00002000	Interrogated by group 13 interrogation
0x00004000	Interrogated by group 14 interrogation
0x00008000	Interrogated by group 15 interrogation

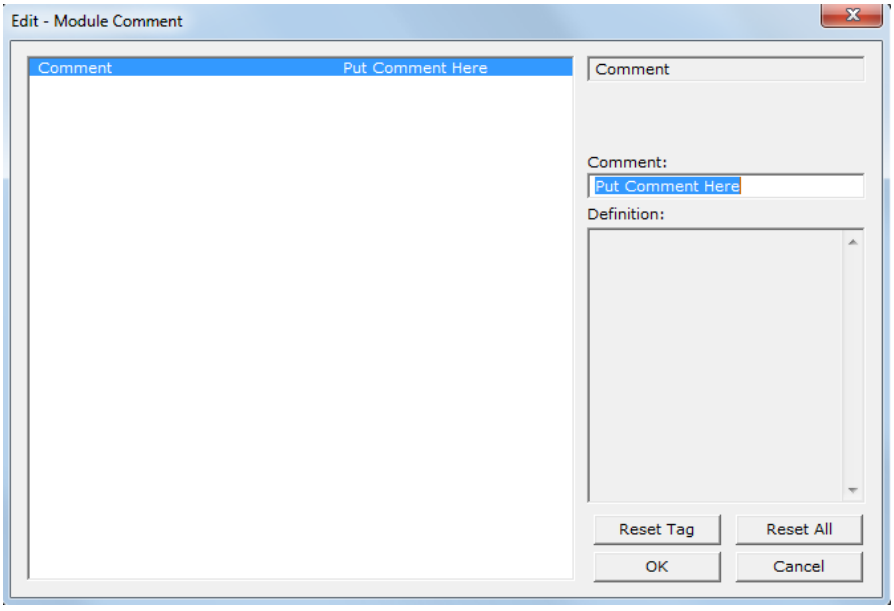
Group Code	Description
0x00010000	Interrogated by group 16 interrogation
0x00020000	Interrogated by general counter request
0x00040000	Interrogated by group 1 counter request
0x00080000	Interrogated by group 2 counter request
0x00100000	Interrogated by group 3 counter request
0x00200000	Interrogated by group 4 counter request
0x40000000	Disable event scanning of this point
0x80000000	Periodic/cyclic data returned from unit

Note 1: Group 3 interrogation also returns the parameter data (ASDU 110 – threshold, low limit and high limit) for all monitored measured points.

If the highest bit (bit 31) is set, data will be produced by the driver for the specified point at the rate set for periodic data generation. Bit 30 (0x40000000) enables scanning of this point for event generation. If the bit is clear and the data type is set for scanning, events will be generated for the point. If the bit is set, events will not be generated for the point. This feature can be used to select which points will generate events for the controlling station and can get rid of event data that is not important to the application.

5.25 To Create Optional Comment Entries

1. Click the plus sign to the left of the Comment icon to expand the module Comments.
2. Double-click the Module Comment icon. The *Edit - Module Comment* dialog appears.



3. Enter your comment and click **OK** to save your changes.

5.26 To print a configuration file

1. Select the module icon, and then click the right mouse button to open a shortcut menu.
2. On the shortcut menu, choose **VIEW CONFIGURATION.** This action opens the *View Configuration* window.
3. On the *View Configuration* window, open the **FILE** menu, and choose **PRINT.** This action opens the *Print* dialog box.
4. On the *Print* dialog box, choose the printer to use from the dropdown list, select printing options, and then click **OK.**

6 Downloading the Project to the Module

In This Chapter

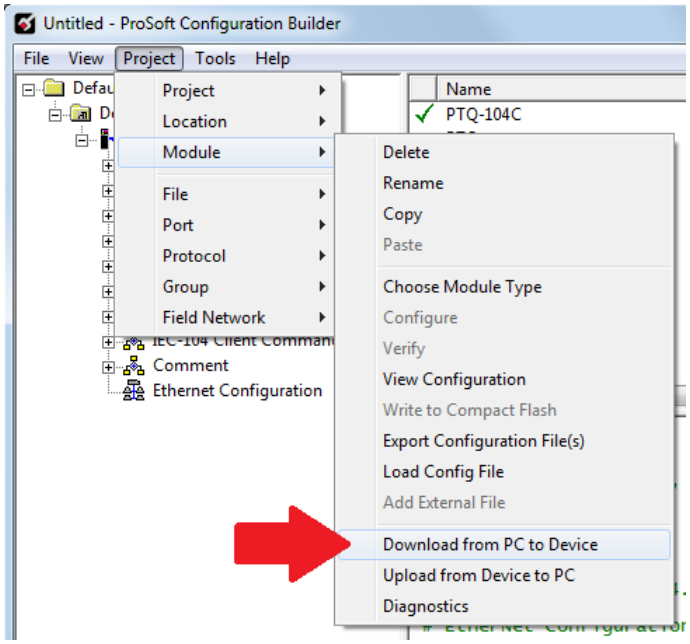
- Downloading via serial connection 102
- Downloading via Ethernet connection 106

6.1 Downloading via Serial Connection

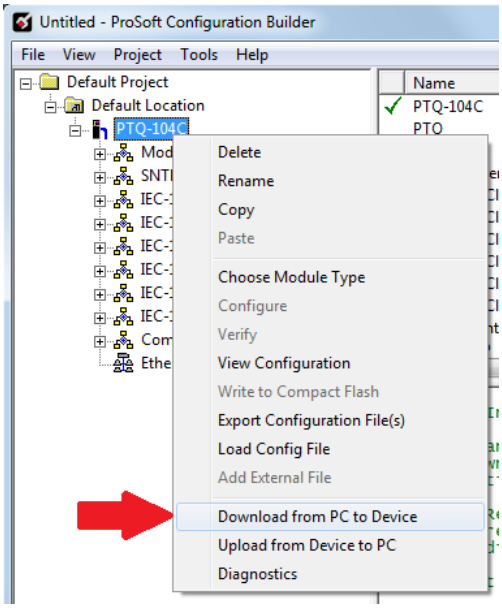
For the module to use the settings you configured, the Project file must be downloaded from your PC to the module.

To Download the Project File

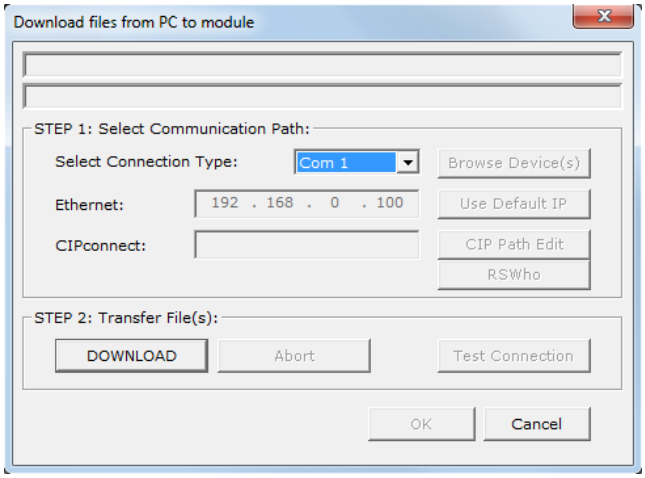
1. In the tree view in ProSoft Configuration Builder, click once to select the PTQ-104S module.
2. Open the **PROJECT** menu from the top of the PCB window, and mouse-over **MODULE.** Select **DOWNLOAD FROM PC TO DEVICE**.



Or, right-click the PTQ-104S icon in PCB and select **DOWNLOAD FROM PC TO DEVICE.**

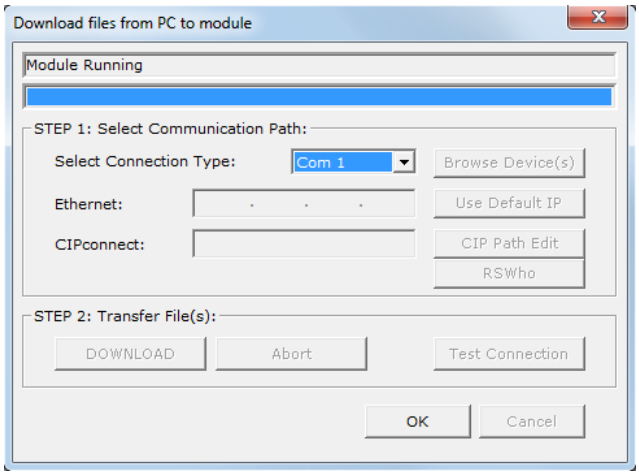


3 The program will scan the PC for a valid com port (this may take a few seconds). When PCB has found a valid com port, the **DOWNLOAD** dialog box will open.



4 Choose the com port to use from the dropdown list, and then click the **DOWNLOAD** button.

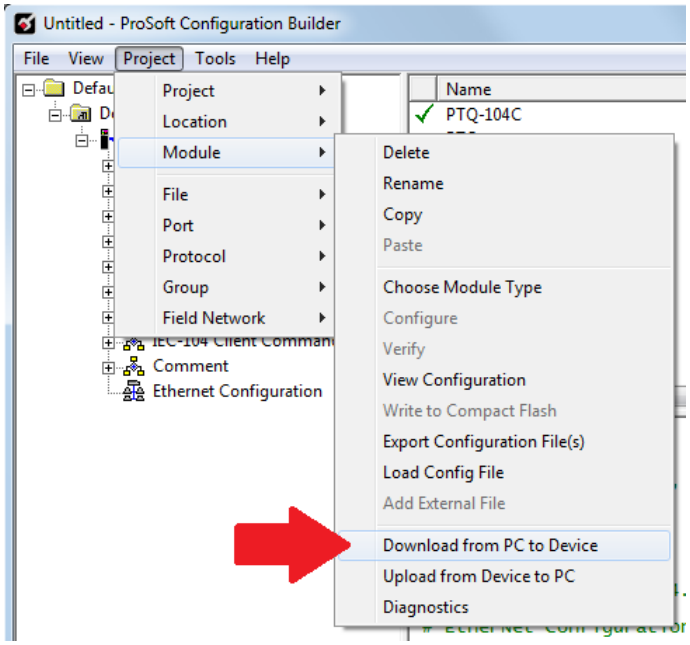
The module will perform a platform check to read and load its new settings. When the platform check is complete, the status bar in the **DOWNLOAD** dialog box will display the message *Module Running*.



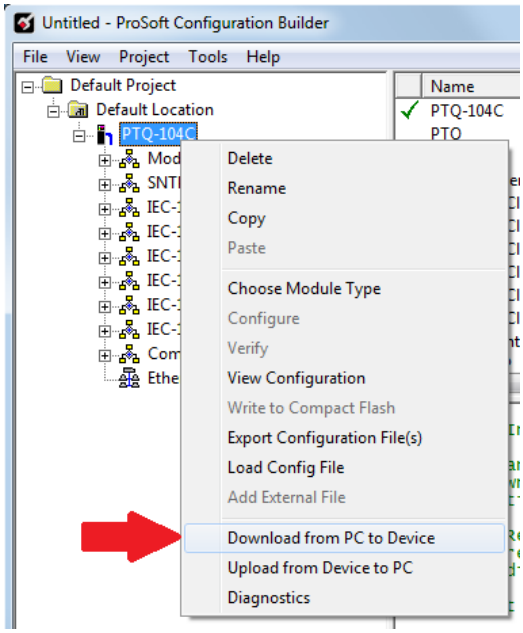
6.2 Downloading the Project via Ethernet Connection

This procedure cannot be done unless the IP address has been initially configured and downloaded serially (page 18) to the module.

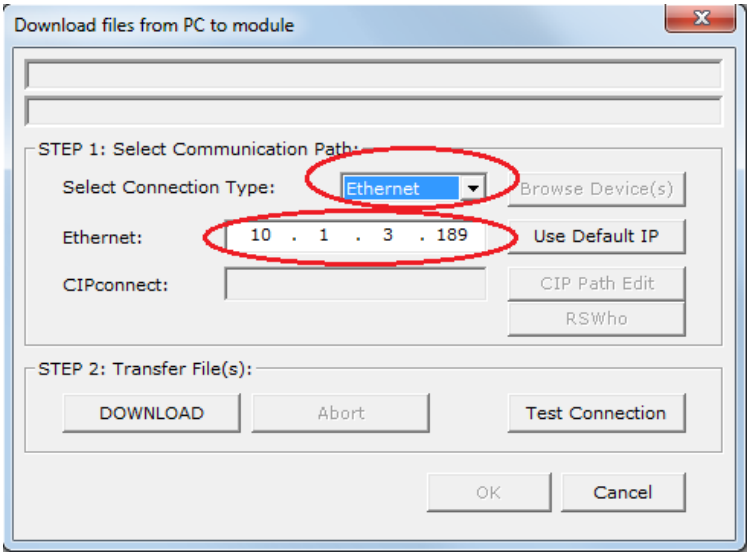
1. In the tree view in ProSoft Configuration Builder, left-click once to select the PTQ-104S module.
2. Open the **PROJECT** menu from the top of the PCB window, and mouse-over **MODULE.** Select **DOWNLOAD FROM PC TO DEVICE**.



Or, right-click the PTQ-104S icon in PCB and select **DOWNLOAD FROM PC TO DEVICE.**

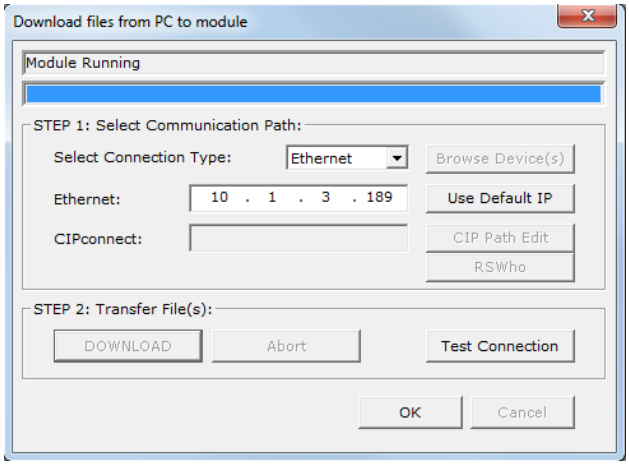


3 The **DOWNLOAD** dialog box will open. Select *Ethernet* as the Connection Type and enter the IP Address that was assigned in the [Ethernet Configuration] section of PCB.



4 Click the **DOWNLOAD** button.

The module will perform a platform check to read and load its new settings. When the platform check is complete, the status bar in the **DOWNLOAD** dialog box will display the message *Module Running*.



7 Hot Standby Support

In This Chapter

- Hot Standby Overview 102
- Setting Up the Modicon Quantum Hot Standby with Unity System 106

7.1 Hot Standby Overview

This section describes the PTQ-104S IEC 60870-5-104 Server module specifications and startup support for Modicon Quantum Hot Standby system.

A Modicon Quantum Hot Standby with Unity system can be used when downtime cannot be tolerated. Hot standby systems deliver high availability through redundancy. A hot standby system consists of two identical configurations.

- Modicon Quantum 140 CPU 671 60
- Modicon Quantum Power Supply Module
- Modicon Quantum RIO Head
- ProSoft Technology PTQ-104S module hardware version 2.05 or higher
- Modicon Optional Modules (NOE, NOM)

One of the 140 CPU 67160s acts as the Primary controller and the other acts as the Standby controller. The Primary controller runs the application program and operates the remote I/O.

Note: The Modicon Quantum RIO Head is required even if the Remote I/O will not be used.

7.1.1 Identical Configurations

Two backplanes are configured with identical hardware and software. One of the programmable logic controllers (PLCs) functions as the Primary controller and the other as a Standby controller, and either controller can be put in the Primary state, but the other must be in the Standby state or offline.

7.1.2 Primary and Standby Controllers

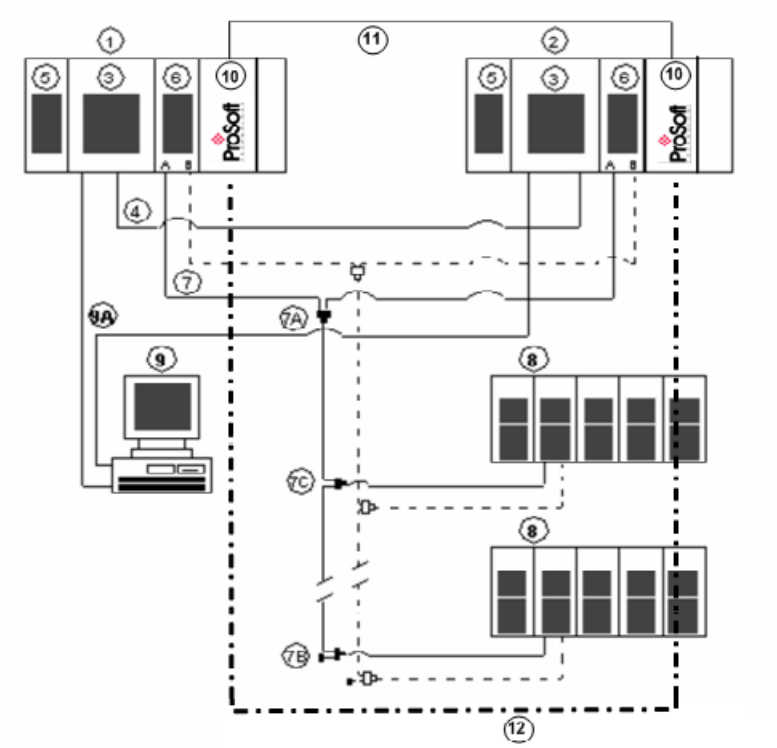
The Primary controller executes the application program, controls the remote I/O, and updates the Standby controller after every scan (program cycle). If the Primary controller fails, the Standby controller takes control within one scan. To determine if the Primary controller failed note controller's status displayed in the HE CPU LCD screen and the RIO Head's status displayed by the RIO Head's LEDs.

The Standby controller does not execute the full application program but only the first section, and the Standby controller does not control the remote I/O but checks out the availability of the Modicon Quantum Hot Standby with Unity equipment.

Note: For additional information on Quantum Hot Standby support, refer to the *Unity Pro Hot Standby User Guide*.

7.1.3 System Components

The following illustration shows the components required for a Modicon Quantum Hot Standby with Unity system.



5 Primary PLC
6 Standby PLC
7 Modicon Quantum Hot Standby with Unity controller with integrated coprocessor
8 Fiber Optic Cable to connect to both controllers
9 Modicon Quantum power supply module: Install power supply in first slot for better rack layout.
10 Modicon Quantum RIO head
11 Coaxial cable with splitters (7A) (MA-0186-100), trunk terminators (7B) (52-0422-000), and tap (7C) (MA-0185-100) for connecting the RIO heads (6) with the RIO drops (8). The dashed connections represent a redundant connection in the RIO network, which is not required for the Modicon Quantum Hot Standby with Unity system.
12 Modicon Quantum RIO drop
13 Unity Pro computer connected to both controllers via Modbus or Modbus Plus (9A)
14 PTQ-104S HSBY modules
15 PTQ-104S Ethernet redundancy communication cable

Note: The 140 CRP 932 00 RIO Head unit is required for Hot Standby System to work.

7.1.4 Modicon Quantum Hot Standby with Unity and IEC Logic

Overview

A Modicon Quantum Hot Standby with Unity system requires two backplanes configured with identical hardware, software, and firmware. One of the controllers (PLC) functions as the Primary controller and the other as a Standby controller.

- The Primary updates the Standby after every scan.
- The Primary and Standby communicate constantly monitoring the health of the system.
- If the Primary fails, the Standby takes control within one scan.

7.1.5 Understanding System Scan Time in Modicon Quantum Hot Standby with Unity Systems

Effect on System Scan Time

The scan time of any Modicon Quantum Hot Standby with Unity system depends on the amount of data transferred. Because data must be transferred from Primary to Standby, any Modicon Quantum Hot Standby with Unity system always has a higher scan time than a comparable stand-alone system.

Performance Considerations

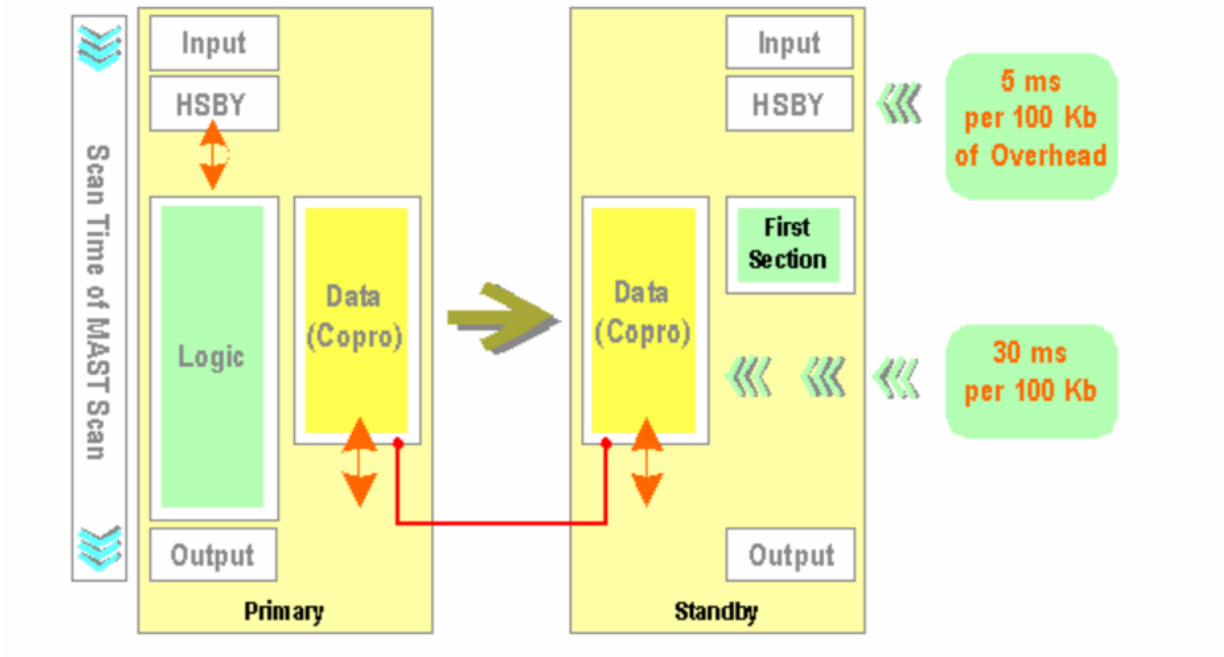
A Modicon Quantum Hot Standby with Unity system increases the length of a MAST scan, creating system overhead.

Note: System overhead is the time required to copy the application data to the communication link layer.

The network scan (communication between Primary and Standby "copros")

1. Exchanges data between both controllers
2. Runs in parallel with the application program.

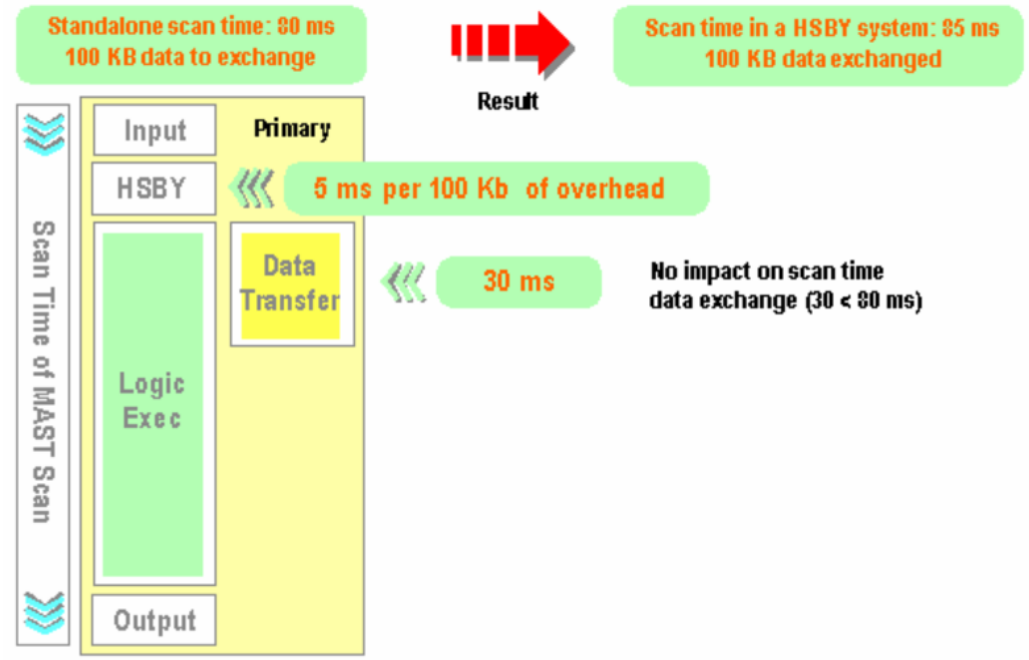
A Hot Standby system



Most of the time, the MAST scan hides the network scan. However, when some application programs are processed, additional system overhead may occur.

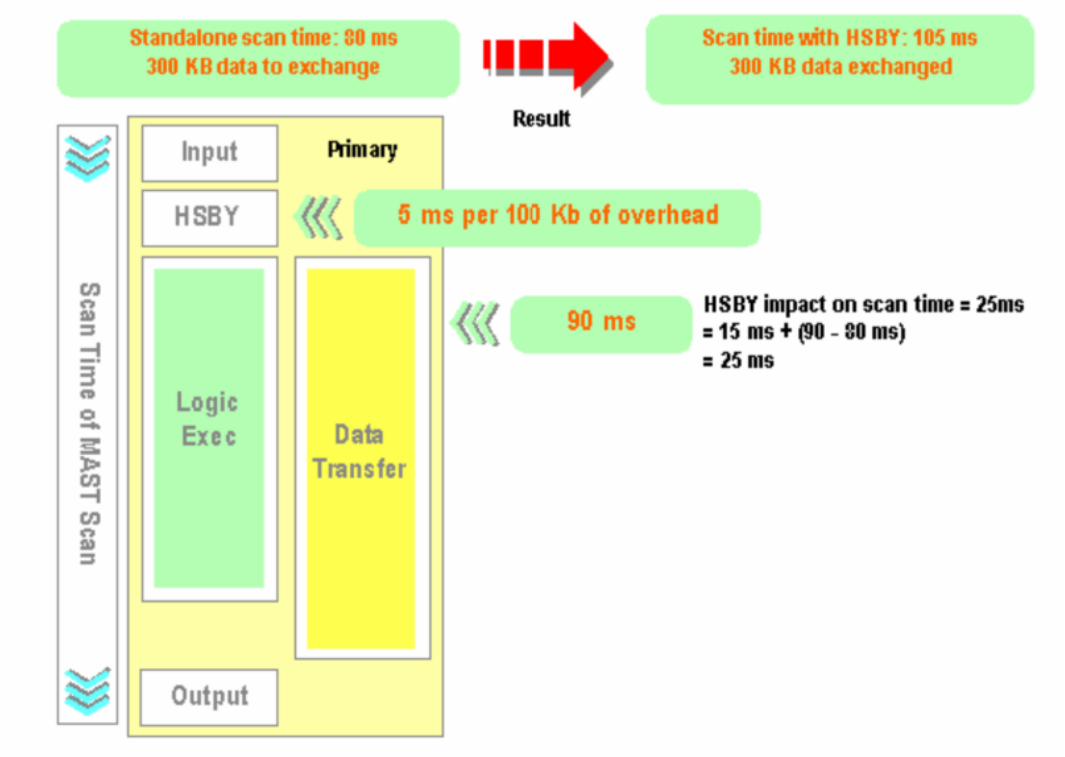
Example #1

- Stand-alone application scan time: 80 ms
- Data (state RAM + unallocated variables): 100 Kb



Example #2

- Stand-alone application scan time: 80 ms
- Data (state RAM + unlocated variables): 300 Kb



Note: In addition to the above times for system overhead, the PTQ-104S module may acquire from 100 ms to 300 ms of switch-over time. All configured data is to be updated as fast as the combined asynchronous events occur based on the processor scan time, backplane transfer time, PTQ data handling time and the 104 Server network update time. The 104 Server update time is based on various other network timings that occur on an Ethernet network.

7.2 Setting Up the Modicon Quantum Hot Standby with Unity System

7.2.1 Overview

Schneider Electric is a leader in offering fault-tolerant, redundant systems, and Hot Standby. Setting up a Modicon Quantum Hot Standby with Unity system involves a number of processes.

7.2.2 Mapping the Backplane Extensions

A Modicon Quantum Hot Standby with Unity requires two backplanes with at least four slots. The two backplanes must be mapped in an identical manner:

- same Modicon Quantum Hot Standby with Unity HE CPU with integrated coprocessor (Copro)
- same firmware
  - same revision level
  - same Modicon Quantum power supply module
  - same Modicon Quantum RIO Head

If other modules are used, for example local I/Os, NOMs, NOEs, these modules must be identical.

For additional information on Modicon Quantum Hot Standby Startup support, refer to the Unity Pro User Guide.

7.2.3 PTQ-104S Hot Standby Considerations

Limitations

The solution allows for up to six PTQ modules per rack (for both Primary and Standby).

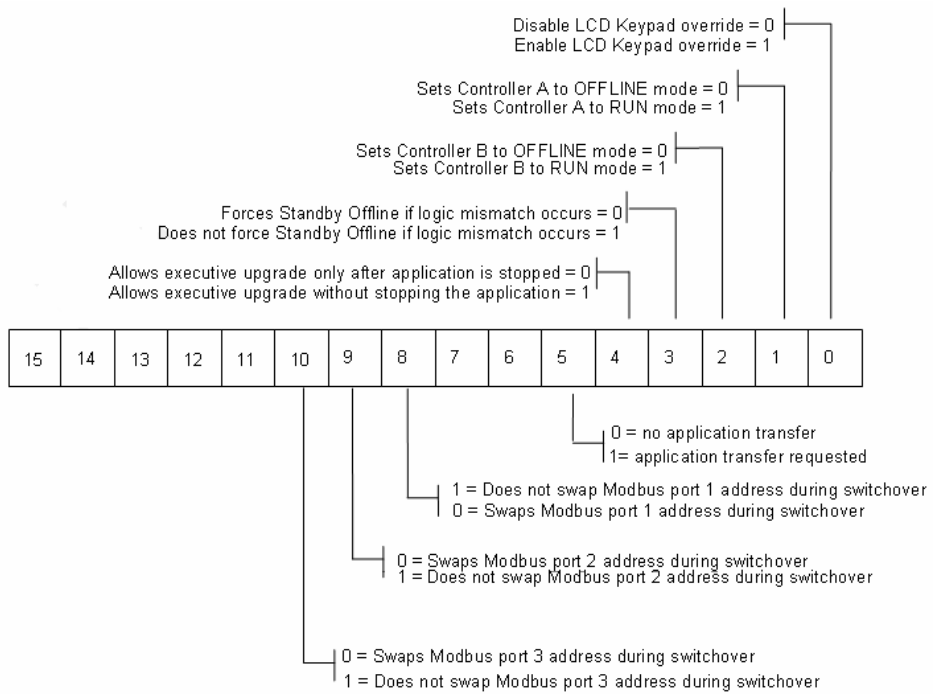
The PTQ-104S module cannot be installed in a RIO drop.

HSBY Operating Modes

Generally, the user will have full control over the switchover via the command register through application program control. This is accomplished by the user application through the SW60 command register.

PLC HSBY Command Register

The following illustration identifies the operating options provided by the Command Register (%SW60). The Command Register defines the operation of the Hot Standby application. That means both the Primary and Standby. Therefore the Command Register is transferred to the Standby PLC each scan. As a result, any changes made to the Command Register on the Standby PLC will have no effect since the value transfer from the Primary side will overwrite it.



%SW60 Hot Standby Command Register

%SW60.0: This bit, if set to 1, allows the Command Register RUN status of the PLC to be set through the LCD Keypad.

Warning: If the keypad override is enabled while the Hot Standby system is running, the Primary PLC will immediately read bits 14 and 15 to determine its own state and the state of the Standby. If both bits are set to 0, a switchover will occur and the former Primary will go offline. The new Primary will continue to operate.

%SW60.1: Setting this bit = 1 will put PLC A in RUN mode. Setting the bit = 0 will put PLC A in OFFLINE mode. This bit takes effect only if bit 16 is set = 1.

%SW60.2: Setting this bit = 1 will put PLC B in RUN mode. Setting the bit = 0 will put PLC B in OFFLINE mode. This bit takes effect only if bit 16 is set = 1.

%SW60.3: Setting this bit = 0 will force the Standby PLC offline if a logic mismatch is detected. Logic mismatch is defined as either the MID, LID or CID being different on Primary and Standby sides. Setting this bit = 1 will allow the Standby PLC to continue to operate normally even if the MID is different on the Primary and Standby.

%SW60.4: Setting this bit = 1 allows the executive to upgrade on the Standby without having to stop the application. This means the Hot Standby system is allowed to operate with different versions of the OS running on the Primary and Standby. This option is provided to allow upgrades to be done without shutting down the process. Clearly, the Standby PLC must be stopped to do the executive upgrade, but it will be able to operate as a valid Standby when started again.

%SW60.5: Setting this bit = 1 commands the standby station to initiate an application transfer. That function is not required in UNITY V1.

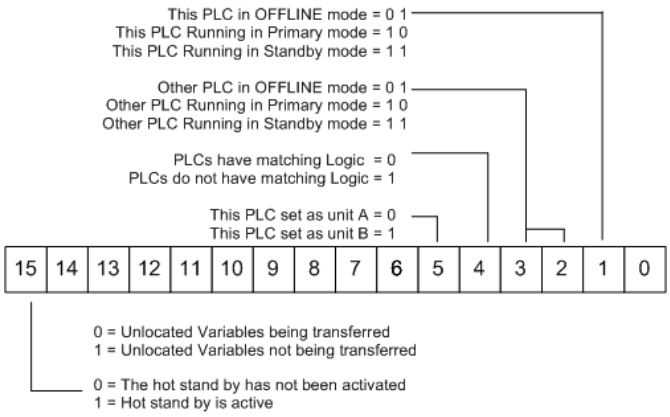
%SW60.8: If this bit is set = 1 the Modbus address on port 1 will be swapped when a switchover occurs. Swapping means to change address by ± 128 to keep the address in the range of 1 to 247. The purpose of this is to allow the P-unit of an HMI to always use the same address to connect to the Primary or Standby/Offline PLC.

PLC HSBY Status Register

The Status Register provides user information relative to the state of the two PLCs in the Hot Standby system. The Status Register is %SW61.

Both the Primary and Standby/Offline PLCs have their own copies of the Status register. The Status register is not transferred from Primary to Standby each scan. Each PLC must maintain its local Status Register based on the regular communication between the two PLCs.

The following illustration identifies the operating options provided by the Command Register.



%SW61 Hot Standby Status Register

The following gives additional detail on the various parts of the Status Register.

%SW61.0 to 3: These bits display the state of the local and remote Hot Standby PLCs.

%SW61.4: This bit is set = 1 whenever a logic mismatch is detected between the Primary and Standby PLCs. This means that either the MID, CID or LID is different on the two PLCs. Under this condition, if bit 13 of the Command Register is set = 0, bit 1 of the Status Register will be set = 1.

%SW61.5: This bit identifies the order reported by the Copro at start time depends on the range of the MAC addresses.

- If the A/B designation is A, then bit 5 will be set = 0.
- If the A/B designation is B, then bit 5 will be set = 1.

%SW61.14: If set = 1 it indicates that a logic mismatch has been detected that disallows Unlocated Variables to be transferred from Primary to Standby. This feature was canceled for UNITY V1.1 because it was determined that a switchover with a partial application context posed too great a hazard.

%SW61.15: If set = 1 it indicates that the Copro device is set up correctly and working.

8 Diagnostics and Troubleshooting

In This Chapter

- The Configuration/Debug Menu 112
- LED Indicators 121

The module provides information on diagnostics and troubleshooting in the following forms:

- LED status indicators on the front of the module provide general information on the module's status.
- Status Data contained in the module can be viewed through the Configuration/Debug port using a terminal emulator program, such as Windows HyperTerminal.
- Status data values can be transferred from the module to processor controller tags and can be monitored there manually or by customer-created logic. For details on Status Data values, see PTQ-104S Status Data Area.

8.1 The Configuration/Debug Menu

The Configuration and Debug menu for this module is arranged as a tree structure, with the *Main* Menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the *Main* menu.

Because this is a text-based menu system, you enter commands by typing the command letter from your computer keyboard in the diagnostic window in ProSoft Configuration Builder (PCB). The module does not respond to mouse movements or clicks. The command executes as soon as you press the **[COMMAND LETTER]** — you do not need to press **[ENTER].** When you type a command letter, a new screen will be displayed in your terminal application.

8.1.1 Required Hardware

You can connect directly from your computer's serial port to the serial port on the module to view configuration information, perform maintenance, and send or receive configuration files.

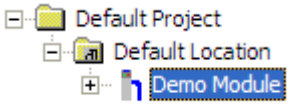
ProSoft Technology recommends the following minimum hardware to connect your computer to the module:

- 80486 based processor (Pentium preferred)
- 1 megabyte of memory
- At least one UART hardware-based serial communications port available. USB-based virtual UART systems (USB to serial port adapters) often do not function reliably, especially during binary file transfers, such as when uploading/downloading configuration files or module firmware upgrades.

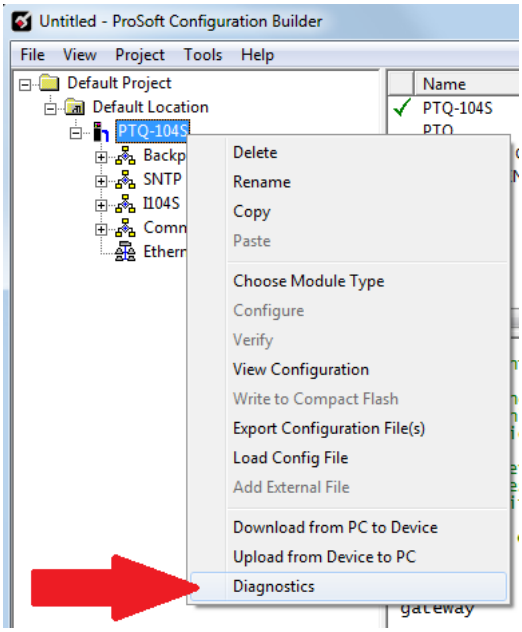
8.1.2 Using the Diagnostic Window in ProSoft Configuration Builder

To connect to the module's Configuration/Debug serial port

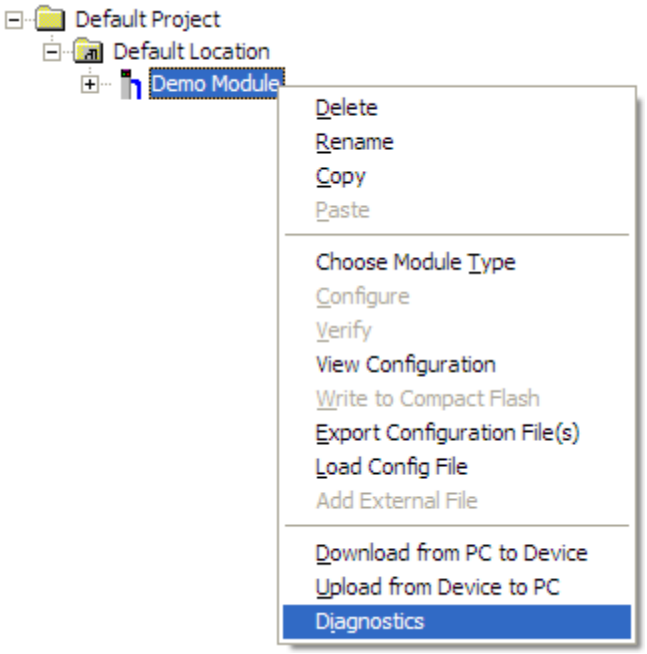
1 Start PCB, and then select the module to test. Click the right mouse button to open a shortcut menu.



Or, if the configuration file has been in use, click the right mouse button on the module icon to open a shortcut menu.

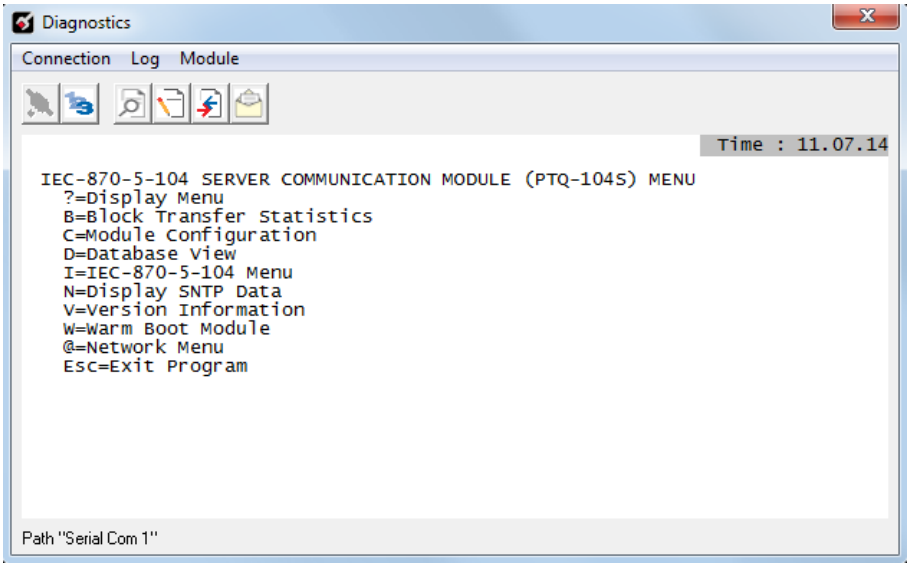


2 On the shortcut menu, choose **DIAGNOSTICS.**



This action opens the **DIAGNOSTICS** dialog box.

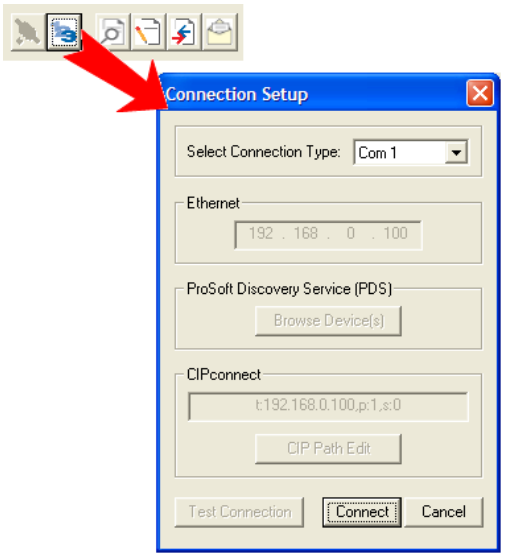
3 Press **[?]** to open the Main Menu.



Important: The illustrations of configuration/debug menus in this section are intended as a general guide, and may not exactly match the configuration/debug menus in your own module.

If there is no response from the module, follow these steps:

1 Click to configure the connection. On the Connection Setup dialog box, select a valid com port or Ethernet connection type.



2 Verify that the null modem cable is connected properly between your computer's serial port and the module. A regular serial cable will not work.

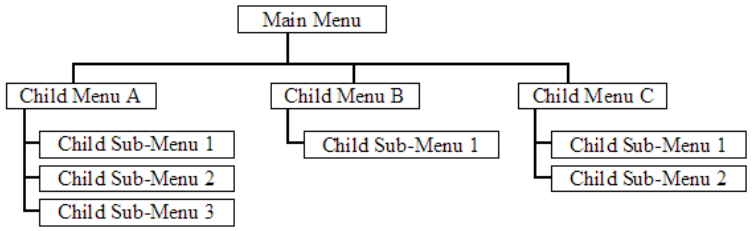
3 On computers with more than one serial port, verify that your communication program is connected to the same port that is connected to the module.

If you are still not able to establish a connection, contact ProSoft Technology for assistance.

8.1.3 Navigation

All of the sub-menus for this module contain commands to redisplay the menu or return to the previous menu. You can always return from a sub-menu to the next higher menu by pressing **[M]** on your keyboard.

The organization of the menu structure is represented in simplified form in the following illustration:



The remainder of this section shows you the menus available for this module, and briefly discusses the commands available to you.

Keystrokes

The keyboard commands on these menus are almost always non-case sensitive. You can enter most commands in lower case or capital letters.

The menus use a few special characters (**[?], [-], [+], [@]**) that must be entered exactly as shown. Some of these characters will require you to use the **[SHIFT], [CTRL]** or **[ALT]** keys to enter them correctly. For example, on US English keyboards, enter the **[?]** command as **[SHIFT][/].**

Also, take care to distinguish capital letter **[I]** from lower case letter **[L]** (L) and number **[1]**. Likewise for capital letter **[O]** and number **[0].** Although these characters look nearly the same on the screen, they perform different actions on the module.

Redisplaying the Menu

Press **[?]** to display the current menu. Use this command when you are looking at a screen of data, and want to view the menu choices available to you.

8.1.4 Main Menu

When you first connect to the module from your computer, your terminal screen will be blank. To activate the main menu, press the **[?]** key on your computer's keyboard. If the module is connected properly, the following menu will appear.

```
IEC-870-5-104 SERVER COMMUNICATION MODULE (PTQ-104S) MENU
  ?=Display Menu
  B=Block Transfer Statistics
  C=Module Configuration
  D=Database View
  I=IEC-870-5-104 Menu
  N=Display SNTP Data
  V=Version Information
  W=Warm Boot Module
  @=Network Menu
  Esc=Exit Program
```

Caution: Some of the commands available to you from this menu are designed for advanced debugging and system testing only, and can cause the module to stop communicating with the processor or with other devices, resulting in potential data loss or other failures. Only use these commands if you are specifically directed to do so by ProSoft Technology Technical Support staff. Some of these command keys are not listed on the menu, but are active nevertheless. Please be careful when pressing keys so that you do not accidentally execute an unwanted command.

Viewing Block Transfer Statistics

Press **[B]** from the Main Menu to view the Block Transfer Statistics screen.

Use this command to display the configuration and statistics of the backplane data transfer operations between the module and the processor. The information on this screen can help determine if there are communication problems between the processor and the module.

Tip: To determine the number of blocks transferred each second, mark the numbers displayed at a specific time. Then some seconds later activate the command again. Subtract the previous numbers from the current numbers and divide by the quantity of seconds passed between the two readings.

Viewing Module Configuration

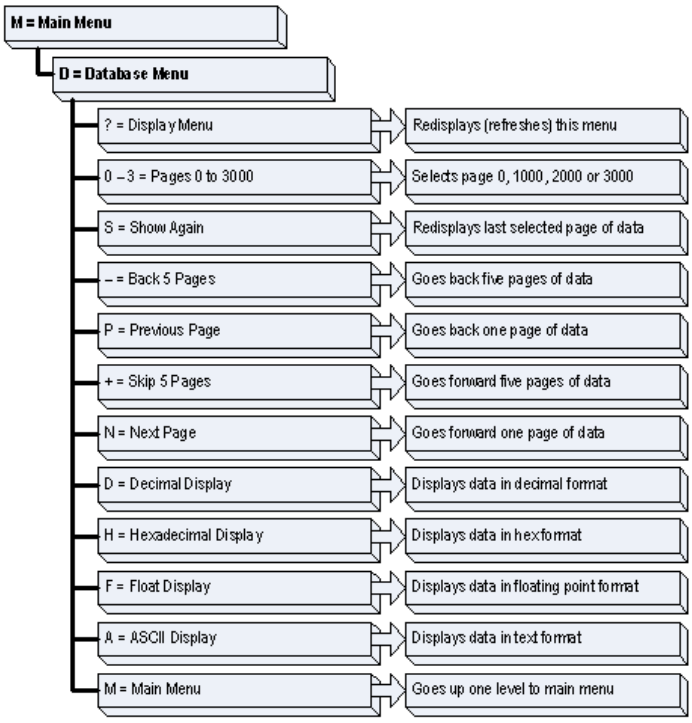
Press **[C]** to view the *Module Configuration* screen.

Use this command to display the current configuration and statistics for the module.

Opening the Database Menu

Press **[D]** to open the Database View menu. Use this menu command to view the current contents of the module's database.

Press **[?]** to view a list of commands available on this menu.



Viewing Register Pages

To view sets of register pages, use the keys described below:

Command	Description
[0]	Display registers 0 to 99
[1]	Display registers 1000 to 1099
[2]	Display registers 2000 to 2099

And so on. The total number of register pages available to view depends on your module's configuration.

Displaying the Current Page of Registers Again

Press **[S]** from the *Database View* menu to show the current page of registers again.

```
DATABASE DISPLAY 0 TO 99 (DECIMAL)
   100   101   102     4     5     6     7     8     9    10
    11    12    13    14    15    16     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
```

This screen displays the current page of 100 registers in the database.

Moving Back Through 5 Pages of Registers

Press **[-]** from the *Database View* menu to skip five pages back in the database to see the previous 100 registers of data start 500 registers before the currently displayed page.

Moving Forward Through 5 Pages of Registers

Press **[+]** from the *Database View* menu to skip five pages ahead in the database to see the next 100 registers of data.

Viewing the Previous 100 Registers of Data

Press **[P]** from the *Database View* menu to display the previous 100 registers of data.

Viewing the Next 100 Registers of Data

Press **[N]** from the *Database View* menu to select and display the next 100 registers of data.

Viewing Data in Decimal Format

Press **[D]** to display the data on the current page in decimal format.

Viewing Data in Hexadecimal Format

Press **[H]** to display the data on the current page in hexadecimal format.

Viewing Data in Floating-Point Format

Press **[F]** from the *Database View* menu. Use this command to display the data on the current page in floating-point format. The program assumes that the values are aligned on even register boundaries. If floating-point values are not aligned as such, they are not displayed properly.

Viewing Data in ASCII (Text) Format

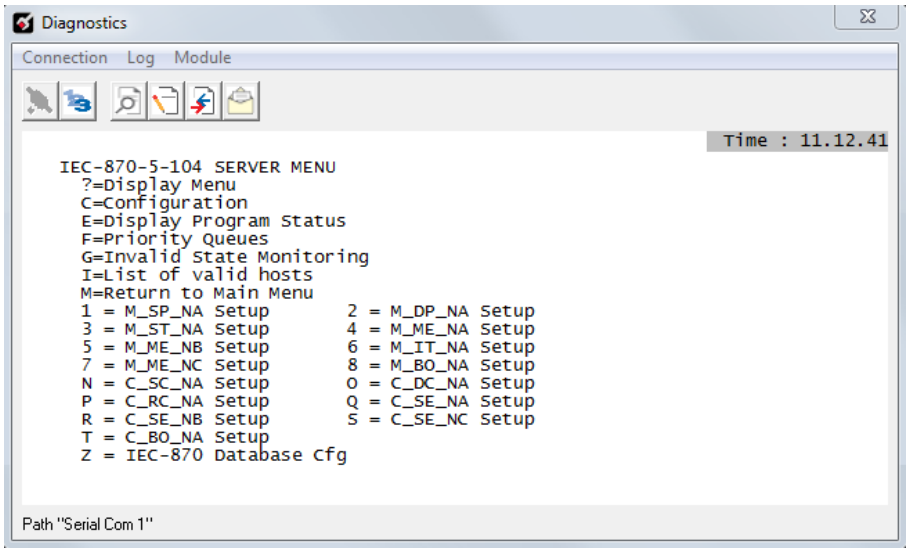
Press **[A]** to display the data on the current page in ASCII format. This is useful for regions of the database that contain ASCII data.

Returning to the Main Menu

Press **[M]** to return to the *Main* Menu.

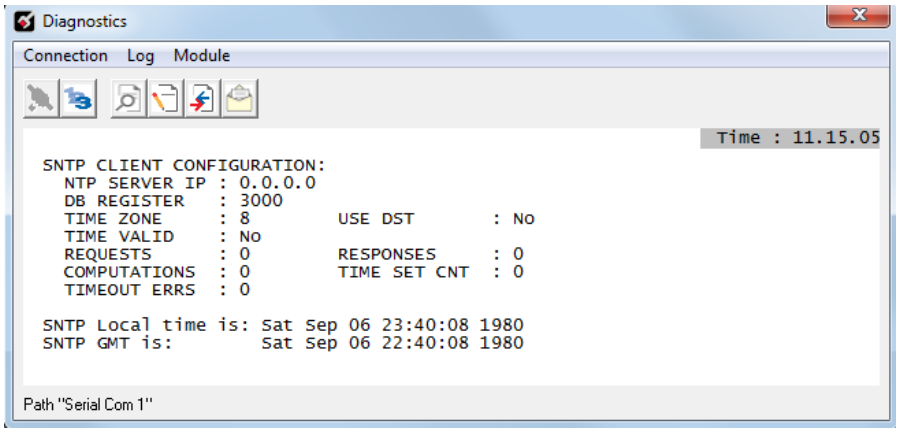
IEC 870-5-104 Menu

Press **[I]** to view all data associated with the IEC 60870-5-104 server driver.



Viewing SNTP Status

Press **[N]** to view configuration information of the SNTP client.



Refer to SNTP Support for more information on configuring and using this function.

Viewing Version Information

Press **[V]** to view Version information for the module.

Use this command to view the current firmware version of the software (*Software Revision Level*) for the module, as well as other important values. You may be asked to provide this information when calling for technical support on the product.

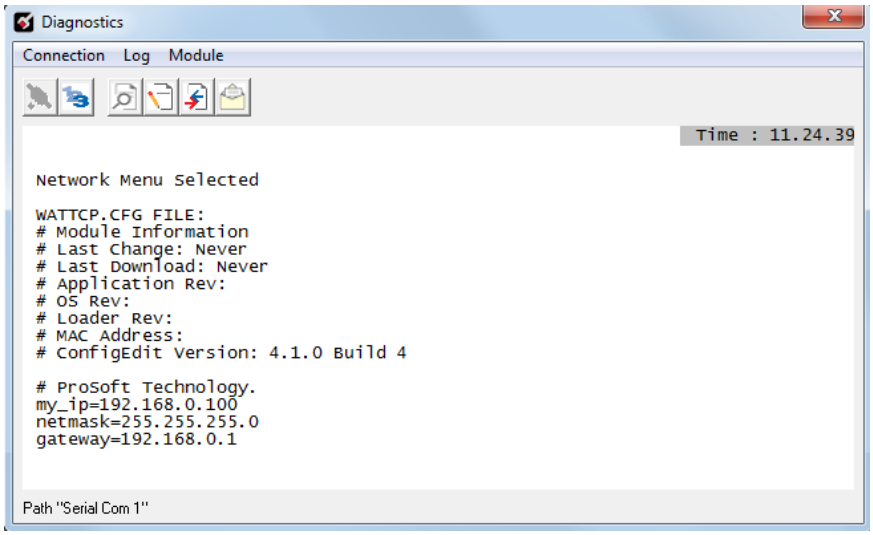
Warm Booting the Module

Caution: Some of the commands available to you from this menu are designed for advanced debugging and system testing only, and can cause the module to stop communicating with the processor or with other devices, resulting in potential data loss or other failures. Only use these commands if you are specifically directed to do so by ProSoft Technology Technical Support staff. Some of these command keys are not listed on the menu, but are active nevertheless. Please be careful when pressing keys so that you do not accidentally execute an unwanted command.

Press **[W]** from the Main Menu to warm boot (restart) the module. This command will cause the program to exit and reload, refreshing configuration parameters that must be set on program initialization. Only use this command if you must force the module to reboot.

Network Menu

Press **[@]** to view the WATTCP.CFG file that contains the IP and module addresses, and other network information. Press **[?]** to view the WATTCP.CFG contents.



Returning to the Main Menu

Press **[M]** to return to the *Main* Menu.

Exiting the Program

Caution: Some of the commands available to you from this menu are designed for advanced debugging and system testing only, and can cause the module to stop communicating with the processor or with other devices, resulting in potential data loss or other failures. Only use these commands if you are specifically directed to do so by ProSoft Technology Technical Support staff. Some of these command keys are not listed on the menu, but are active nevertheless. Please be careful when pressing keys so that you do not accidentally execute an unwanted command.

Press **[Esc]** to restart the module and force all drivers to be loaded. The module will use the configuration stored in the module's Flash memory to configure the module.

8.2 LED Indicators

The LEDs indicate the module's operating status as follows:

LED	Color	Status	Indication
PRT1	Green	On	Data is being transferred between the module and a remote terminal using the Configuration/Debug port.
		Off	No data is being transferred on the Configuration/Debug port.
PRT2	Green	On	Port not used in application
		Off	Port not used in application
PRT3	Green	On	Port not used in application
		Off	Port not used in application
ERR1	Red	Off	The PTQ-104S is working normally.
		On	The PTQ-104S module program has recognized an application error.
ERR2	N/A		Not used in application
ERR3	Red	On	Configuration Error
Active	Green	On	The LED is on when the module recognizes a processor and is able to communicate if the [Backplane Configuration] section specifies data transfer commands.
		Off	The LED is off when the module is unable to speak with the processor. The processor either absent or not running.
BAT	Red	Off	The battery voltage is OK and functioning.
		On	The battery voltage is low or the battery is not present. The battery LED will illuminate briefly upon the first installation of the module or if the unit has been un-powered for an extended period of time. This behavior is normal; however should the LED come on in a working installation please contact ProSoft Technology.

If your module is not operating, and the status LEDs are not illustrated in the table above, please call ProSoft Technology for technical assistance.

8.2.1 Ethernet LED Indicators

LED	State	Description
Data	Off	No activity on the Ethernet port.
	Green Flash	The Ethernet port is actively transmitting or receiving data.
Link	Off	No physical network connection is detected. No Ethernet communication is possible. Check wiring and cables.
	Green Solid	Physical network connection detected. This LED must be on solid for Ethernet communication to be possible.

8.2.2 Error Status Table

The program maintains an error/status table that is transferred to the processor in each read block. You can use the error/status data to determine the "health" of the module. Refer to Error Status Table (page 171) for data block structure.

9 Reference

In This Chapter

- Product Specifications 124
- PTQ-104S Protocol Implementation 127
- Cable Connections 170
- Error Status Table 171
- Group Codes 174
- IEC 60870-5-104 Server Interoperability Statement 175
- PTQ-104S Database Design Forms 186
- Frequently Asked Questions 190

9.1 Product Specifications

9.1.1 Standards

The standards used in developing the product are listed in the following table.

PUBLICATION	TITLE
IEC 60870-5-104	Companion Standard for Basic Telecontrol Tasks
IEC 60870-5-104 Amendment 1	Companion Standard for Basic Telecontrol Tasks
IEC 60870-5-1	Transmission Frame Formats
IEC 60870-5-2	Link Transmission Procedures
IEC 60870-5-3	General Structure of Application Data
IEC 60870-5-4	Definition and Coding of Application Information Elements
IEC 60870-5-5	Basic Application Functions

The PTQ IEC 60870-5-104 Server Communication Module allows Quantum compatible processors to interface easily with IEC 60870-5-104 protocol-compatible hosts.

The PTQ-104S module acts as an input/output module between the IEC-60870-5-104 Ethernet network and the Quantum processor. Data transfer between the module and the processor is asynchronous from the actions on the network. Databases are defined by the user for the module to hold data as required by the protocol.

The PTQ-104S module is a powerful communication interface for Quantum processors. Developed under license from Schneider Electric, the module incorporates proprietary backplane technology that enables powerful data access to the Quantum processor.

9.1.2 General Specifications

- Single Slot - Quantum backplane compatible
- The module is recognized as an Options module and has access to PLC memory for data transfer
- Configuration data is stored in non-volatile memory in the ProTalk® module
- Up to six modules can be placed in a rack
- Local rack - The module must be placed in the same rack as processor
- Compatible with all common Quantum programming packages, including Concept (version 2.6 or higher), Unity Pro (version 2.2 or higher), ProWORX (version 2.20 or later), and ModSoft
- Quantum data types supported: 3x, 4x
- High speed data transfer across backplane provides quick data update times
- Supports Hot-Standby processors or applications

9.1.3 Hardware Specifications

Specification	Value
Backplane Current Load	1100 mA maximum @ 5 Vdc ± 5%
Operating Temperature	0°C to 60°C (32°F to 140°F)
Storage Temperature	-40°C to 85°C (-40°F to 185°F)
Relative Humidity	5% to 95% (without condensation)
Vibration	Sine vibration 4-100 Hz in each of the 3 orthogonal axes
Shock	30G, 11 mSec. in each of the 3 orthogonal axes
Dimensions (HxWxD), Approx.	250 x 40.34 x 103.85 mm 9.84 x 1.59 x 4.09 in
LED Indicators	Module Status Backplane Transfer Status Serial Port Activity LED Serial Activity and Error LED Status
Debug/Configuration Port (Debug)	
CFG Port (DEBUG)	DB-9M PC Compatible RS-232 only No hardware handshaking
Application Ports	
Application Ethernet port	RJ45 Connector Link and Activity LED indicators Electrical Isolation 1500 V rms at 50 Hz to 60 Hz for 60 s, applied as specified in section 5.3.2 of IEC 60950: 1991 Ethernet Broadcast Storm Resiliency = less than or equal to 5000 [ARP] frames-per-second and less than or equal to 5 minutes duration

9.1.4 Functional Specifications

The PTQ-104S module accepts data read/write commands from a master/client on the network. In addition, the module can be configured to generate unsolicited messages in either a spontaneous or cyclic fashion.

The module has 4000 words of user defined internal register space that are accessible to the protocol driver and to the Quantum processor memory. Any of the supported database types can be individually located (within the total database size limit of 4000 words) and each database point is mapped within the module and can be assigned to one or more Groups.

The supported database point types are:

Database Type	Description	Event Generation	Point Size
M_SP_NA	Monitored single-point database	State Change	1 bit
M_DP_NA	Monitored dual-point database	State Change	2 bits
M_ST_NA	Monitored step-point database	State Change	1 byte
M_ME_NA	Monitored normalized-point database	Dead-band	1 word
M_ME_NB	Monitored scaled-point database	Dead-band	1 word
M_ME_NC	Monitored short-float point database	Dead-band	2 words
M_IT_NA	Monitored integrated total database	N/A	2 words
C_SC_NA	Command single-point database	N/A	1 bit
C_DC_NA	Command dual-point database	N/A	2 bit
C_RC_NA	Command step-point database	N/A	1 byte
C_SE_NA	Command normalized-point database	N/A	1 word
C_SE_NB	Command scaled-point database	N/A	1 word
C_SE_NC	Command short-float point database	N/A	2 words

IEC 60870-5-104 Server Specifications

Operating in the Server mode, the module accepts commands from a Client(s) to read/write data stored in the module's internal registers. This data is easily and continuously transferred between the ProTalk module and the Quantum processor's data registers.

The IEC 60870-5-104 Server functionality supported by the module includes:

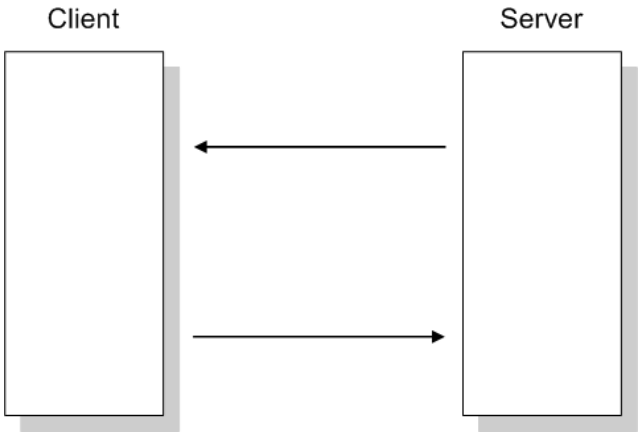
- The IEC 60870-5-104 communication driver is built in accordance to the approved IEC specification
- The module functions as a Server on the network supporting data read/write commands from an IEC 60870-5-104 client on the network
- One TCP server socket
- Supports unsolicited messages, either cyclic or spontaneous (cyclic rate is configurable)
- Supports clock synchronization commands from client or from the Quantum
- Supports Group interrogation (Global, General 1 to 16, Global Counters, Counters 1 to 4)
- Event time-stamping configurable by type (None, 24 bit, 56 bit)
- Event queue supports 99 points for each data type
- Optional user defined list of acceptable client host IP addresses
- Configurable Common ASDU address (sector) and Information Object Address
- An IEC Interoperability Document for the ProTalk is available which fully documents data types supported by the module

9.2 PTQ-104S Protocol Implementation

The intent of this section is to provide a quick understanding of how the PTQ-104S module implements the IEC-60870-5-104 protocol, without going into complex details of the specification.

The IEC-60870-5-104 protocol applies to Telecontrol equipment and systems with data transmission for monitoring and controlling geographically widespread processes. This protocol consists essentially of the IEC-60870-5-101 protocol, with the addition of TCP/IP as the transport mechanism.

Any application with the IEC-60870-5-104 protocol consists of a client (Controlling Station) and one or more servers (Controlled Stations). The client constantly monitors and controls the data from each server in the TCP/IP network.



The PTQ-104S works as an IEC-60870-5-104 server; it can send monitor data, receive commands, or generate events to the client unit.

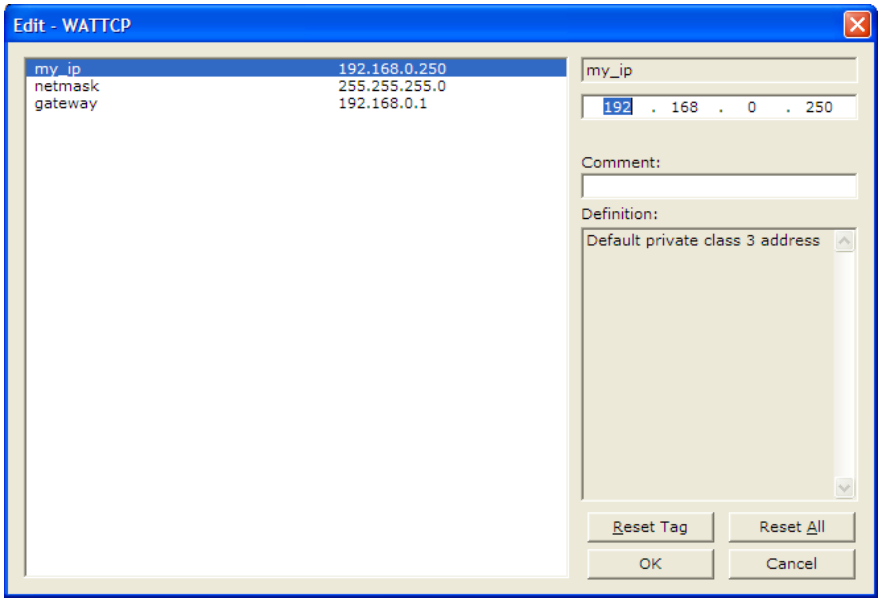
9.2.1 Module Address

The PTQ-104S module is identified at transport level using the IP Address and at application level using the Common ASDU Address.

IP Address

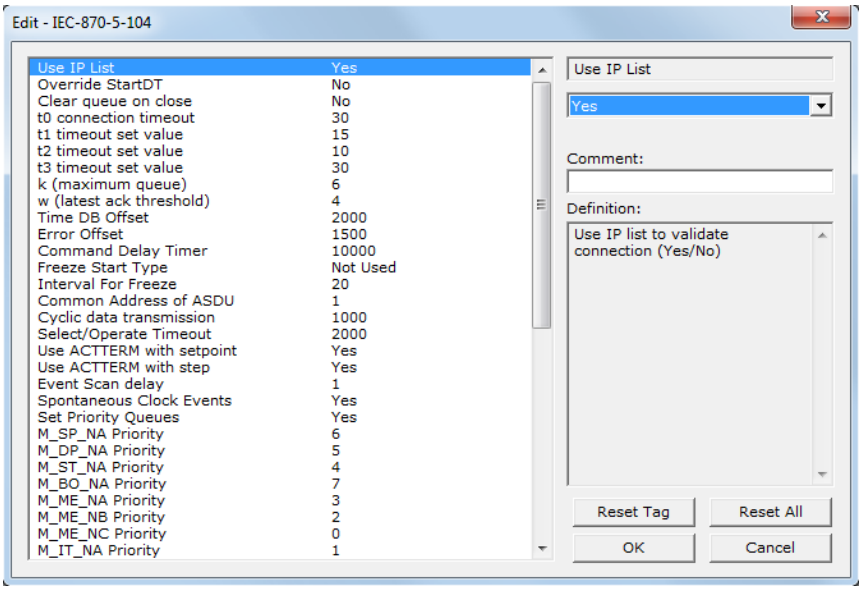
The PTQ-104S module is identified by a unique IP address on the TCP/IP network. You must edit the WATTCP.CFG configuration file (or use the configuration tool) to enter a valid IP address.

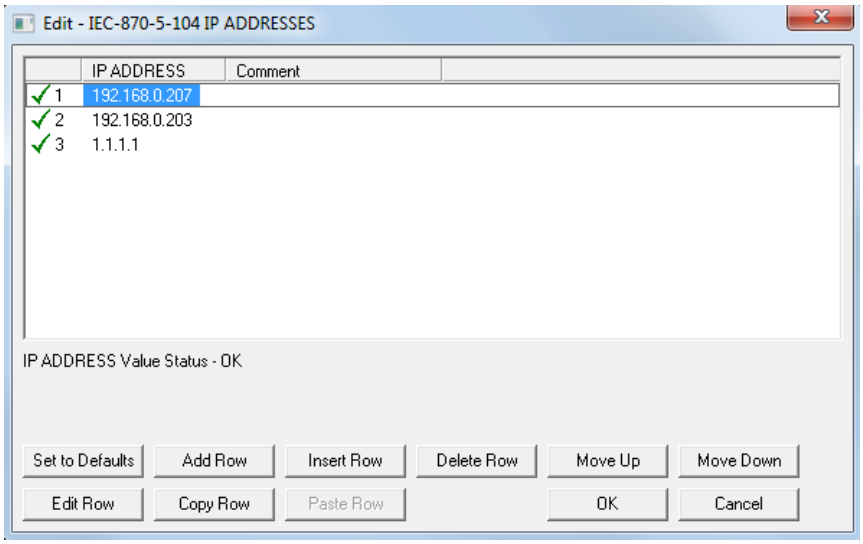
The following example lists the default contents of the WATTCP.CFG file:



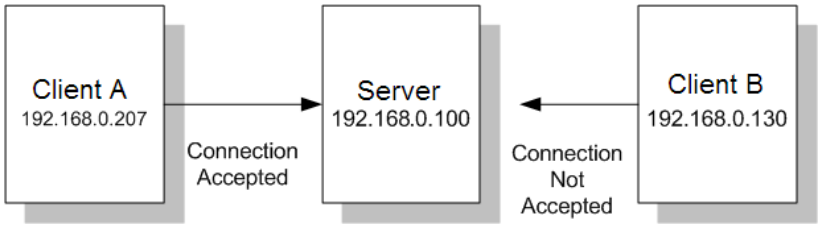
In this example, the PTQ-104S module is identified by IP address 192.168.0.250 in the IEC-60870-5-104 network, with a netmask (subnet mask) of 255.255.255.0 and a default gateway address of 192.168.0.1.

Because there could be several devices in the same TCP/IP network, some applications may require a connection control (from which IP addresses the module may receive valid messages.) To restrict the units (IP addresses) from which the module will accept connections, enable the *Use IP List* parameter.





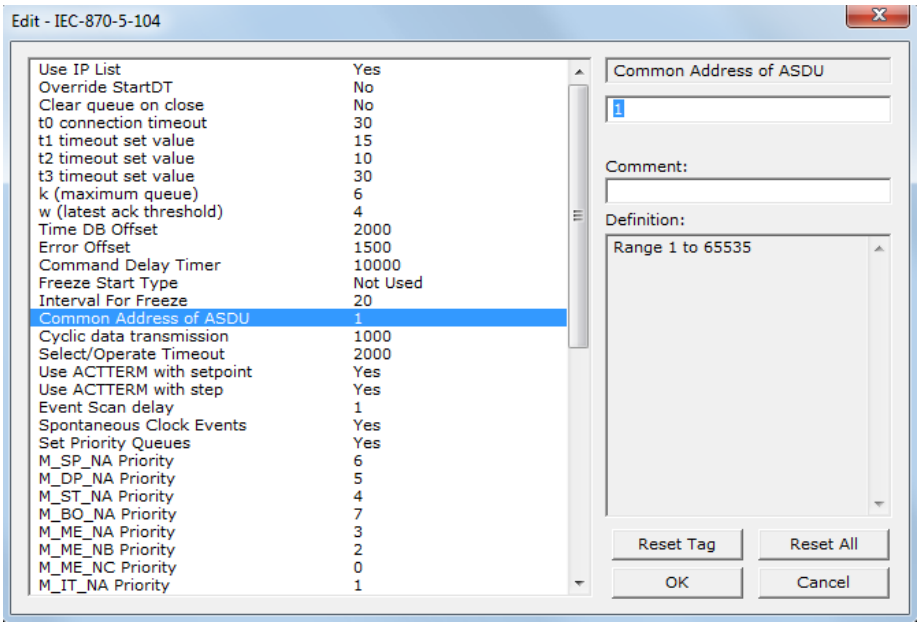
If the *Use IP List* parameter is set to **YES**, the module will only accept a connection from one of the three IP addresses listed, as in the example above. The following illustration shows that the Server will accept a connection from Client A, whose address is on the list, but will reject a connection from Client B, whose address is not on the list.



Common ASDU Address

At the application level, the module is identified by the Common ASDU (Application Service Data Unit) Address. This address must match the CASDU sent by the Client unit. An ASDU is a data unit that transfers information objects between the Client and the server.

If the Client sends a message to a different Common ASDU, the module ignores the command. To configure the Common ASDU Address for the PTQ-104S, use the *Common Address of ASDU* parameter in the configuration file.

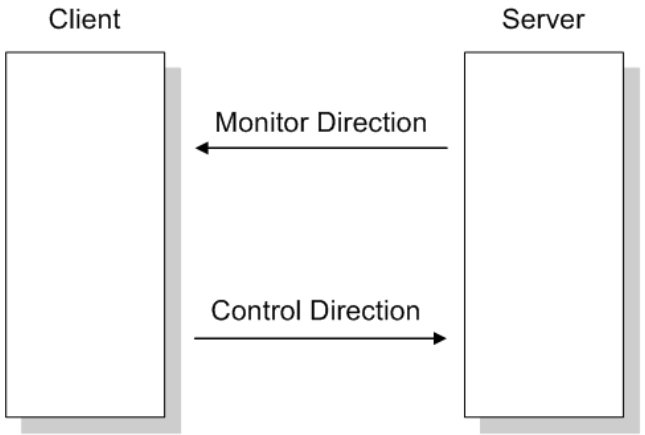


9.2.2 Monitor Direction and Control Direction: Point Definition

The protocol specification defines two directions of data: monitor direction and control direction.

Monitor Direction: The direction of transmission from a server to a Client

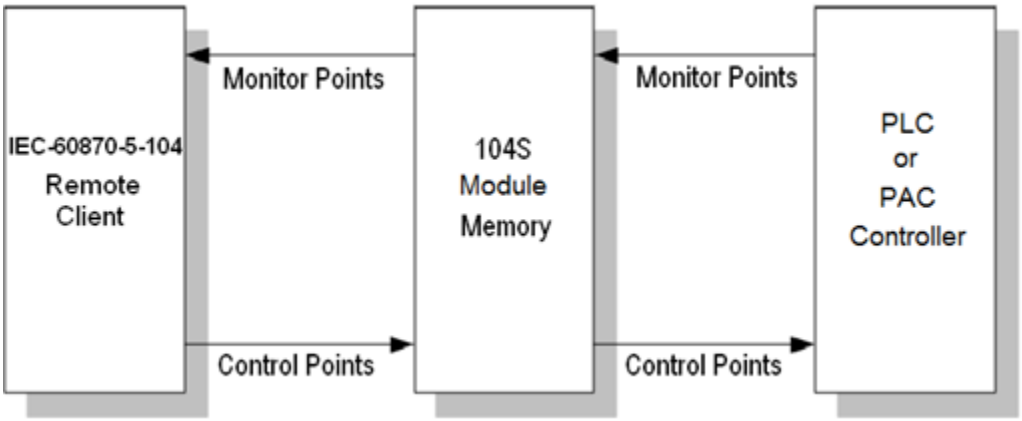
Control Direction: The direction of transmission from a Client to a server



The points that are typically transferred from a server to a Client are also known as **Monitor Points** (or Monitor Information Objects). The points that are typically transferred from a Client to a server are also known as **Control Points** (or Command Information Objects).

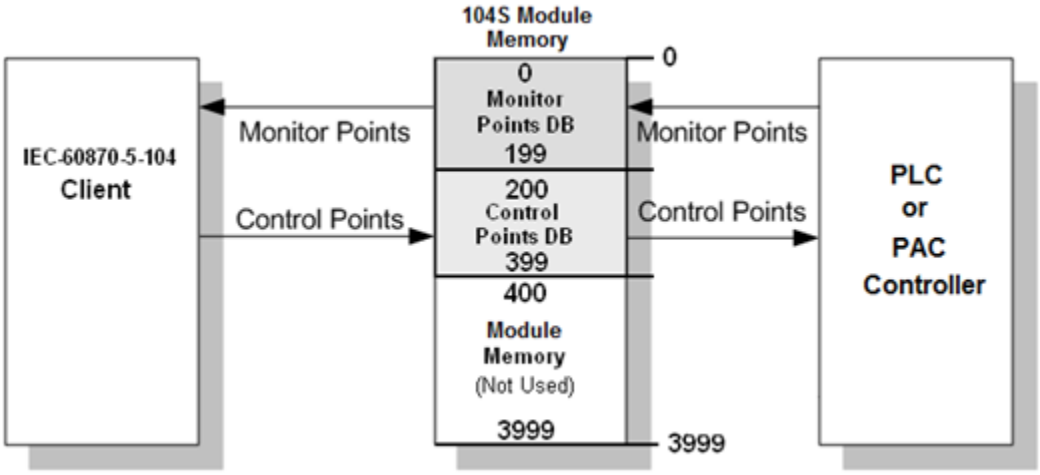
The PTQ-104S contains an internal database of 4000 16-bit words. You must associate the monitor and control points to database addresses in the PTQ-104S. To configure the points for the PTQ-104S, follow these steps:

1. Calculate the number of monitor and control points for the application.
2. Calculate the PTQ-104S database regions that are required for the application, based on the number of monitor and control points. Define two separate regions. Remember that each data type stores a different quantity of data (for example, M_SP_NA uses one bit, M_ST_NA uses one byte, and so on).
3. Configure each point within its PTQ-104S database region.
4. Make sure that the other parts of your application correctly update the module database regions associated with the configured 104S datatypes, as shown in the following illustration.



All points must be configured in the correct location in the PTQ-104S database in order to be properly updated by other parts of the application. Keep the datatypes separated by configuring the control points and monitor points in separate areas of the PTQ-104S database.

The following illustration shows an example configuration:



In this example, all monitor points are located between database addresses 0 and 199, and all control points are located between address 200 and 399.

9.2.3 Using Monitor Points

The following monitor points are supported by the PTQ-104S module:

M-SP-NA	Monitored Single-Points	1 bit	Bit
M-DP-NA	Monitored Dual-Points	2 bits	Bit
M-ST-NA	Monitored Step-Points	1 byte	Byte
M-ME-NA	Monitored Measured Normalized-Points	1 word	Word
M-ME-NB	Monitored Measured Scaled-Points	1 word	Word
M-ME-NC	Monitored Measured Short Floating Points	2 words	Double-Word
M-IT-NA	Monitored Counter-Points	2 words	Double-Word

Each monitor point is identified by its Information Object Address (it should be unique for each Common ASDU Address in the network). For each monitor point, configure the following parameters:

Point # - The information object address of the point. It identifies the point in the network.

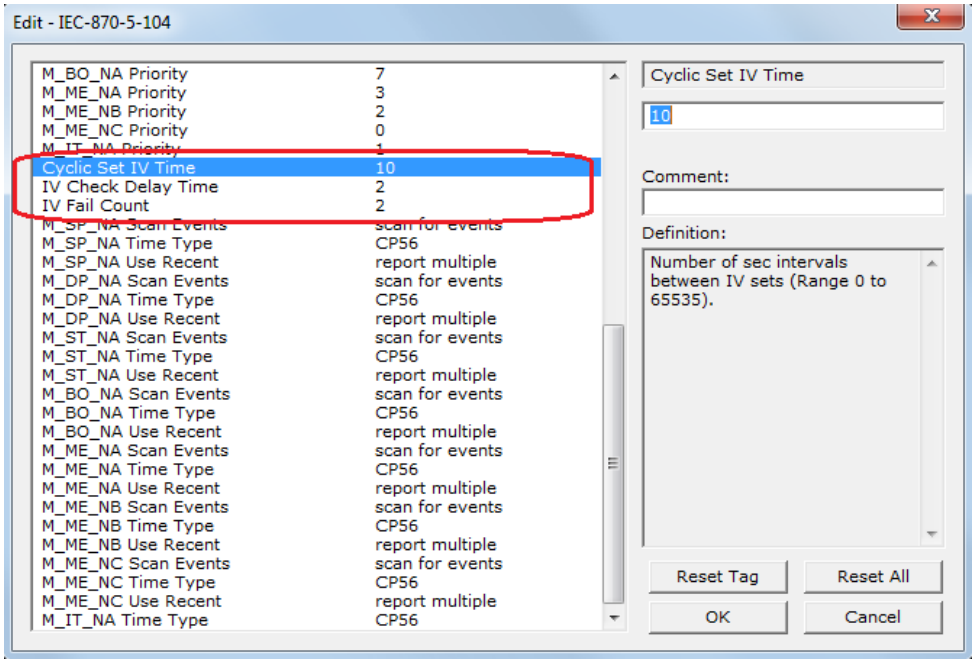
DB Address - The database location in the PTQ-104S module associated with the point. You must associate each point to a database address in the PTQ-104S module. The interpretation of this parameter depends on the point type configured. For example, for an M_SP_NA point, this value represents the bit address. For a M_ME_NA point, this value represents the Word address.

Group(s) - This is the group definition for the point. It sets how the point will be polled by the Master (cyclic or group interrogation). It can also be used to enable or disable the event generation for one specific point. The group parameter is discussed in the Data Communication section.

Deadband - Sets the deadband for each measured point. If the value changes from more than the configured deadband, the module will generate an event for this point.

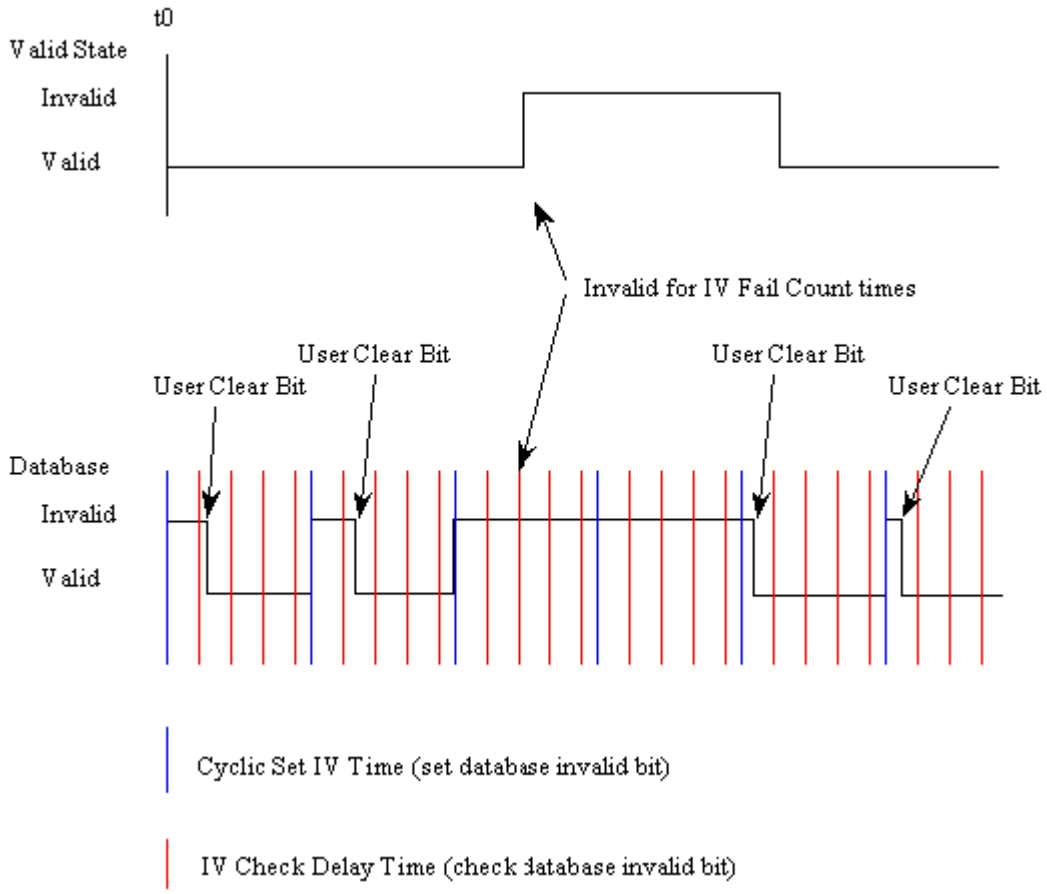
IV DB Bit - This feature allows the application to set the invalid (IV) quality bit of the protocol for all the monitored ASDU types supported. If you enable this feature, the processor can determine the individual IV quality bit status of each point you configured.

The following parameters must be configured in order to use this feature:



To disable this feature, set the IV Fail Count parameter to 0. If used, the Cyclic Set IV Time parameter must be at least 3 times larger than the IV Check Delay Time.

The Cyclic Set IV Time parameter must be set to determine how frequently the IV Checks will be performed. If the IV bit is ON for a number of times given by the IV Fail Count parameter the module will consider the point as invalid. The following illustration shows how these parameters are implemented:



If the IV bit field is absent or set to 0, the invalid quality state for the point will always be reported as valid.

If a database bit address (1 to 64000) is present, the application may consider the point with an invalid flag if the previous logic checks the IV bit as 1 during consecutive IV Check Delay scans. The IV bits would have to be reset to 0 to set the point to valid state.

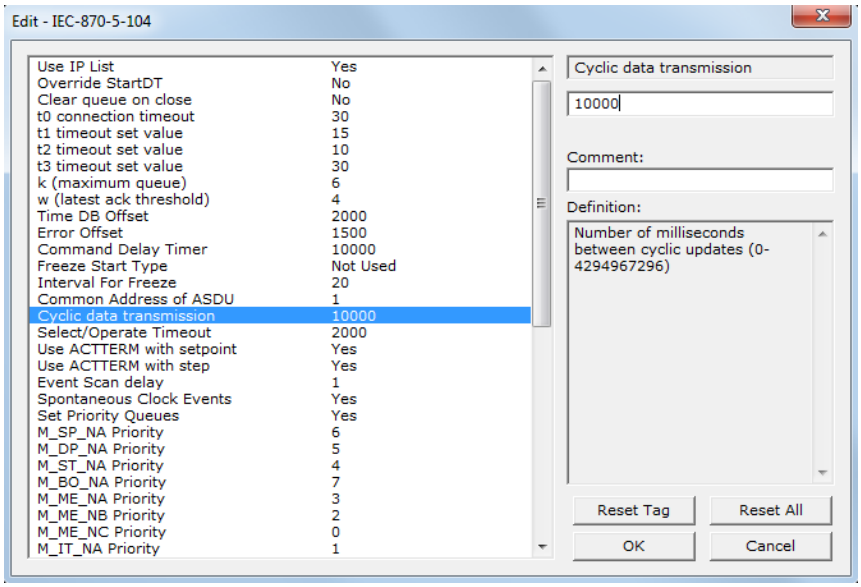
The IV DB bit defined for each point can be unique or many points may share the same bit. The last case could be used when the points on an I/O module are to be considered as one set. In this case only a single bit is required. For a point that is the result of a computation, the valid quality state could be set for each point individually.

Monitor Data Transfer

Typically, you should properly configure the group code for each monitor point to define how the Master will poll for the point. The group codes are defined as follows:

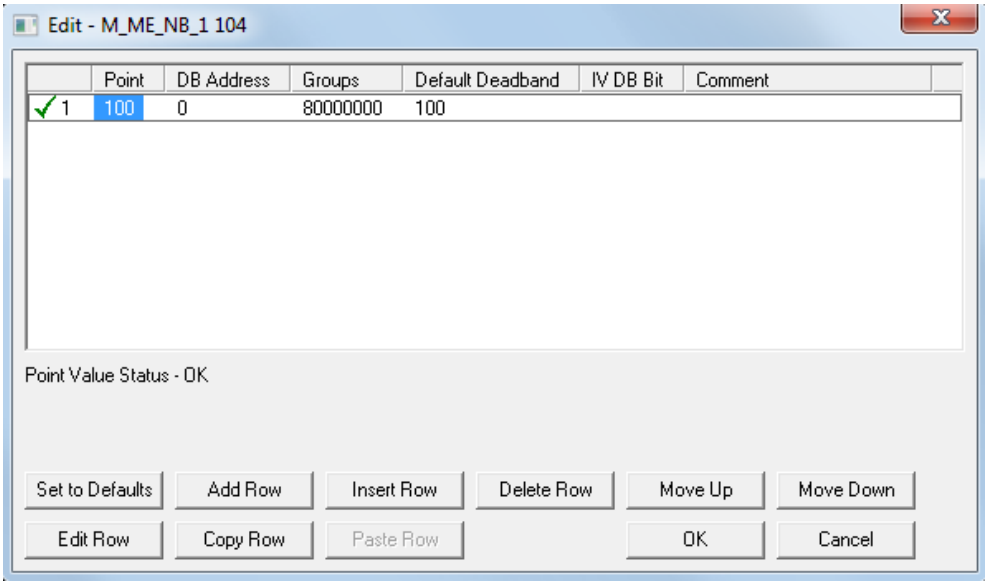
Group Code	Description
0x00000001	Interrogated by general interrogation (station or global)
0x00000002	Interrogated by group 1 interrogation
0x00000004	Interrogated by group 2 interrogation
0x00000008	Interrogated by group 3 interrogation
0x00000010	Interrogated by group 4 interrogation
0x00000020	Interrogated by group 5 interrogation
0x00000040	Interrogated by group 6 interrogation
0x00000080	Interrogated by group 7 interrogation
0x00000100	Interrogated by group 8 interrogation
0x00000200	Interrogated by group 9 interrogation
0x00000400	Interrogated by group 10 interrogation
0x00000800	Interrogated by group 11 interrogation
0x00001000	Interrogated by group 12 interrogation
0x00002000	Interrogated by group 13 interrogation
0x00004000	Interrogated by group 14 interrogation
0x00008000	Interrogated by group 15 interrogation
0x00010000	Interrogated by group 16 interrogation
0x00020000	Interrogated by general counter request
0x00040000	Interrogated by group 1 counter request
0x00080000	Interrogated by group 2 counter request
0x00100000	Interrogated by group 3 counter request
0x00200000	Interrogated by group 4 counter request
0x40000000	Disable event scanning of this point
0x80000000	Periodic/cyclic data returned from unit

The module will periodically send all points configured for periodic/cyclic poll (0x80000000) at every x milliseconds, where x is configured with the following parameter:

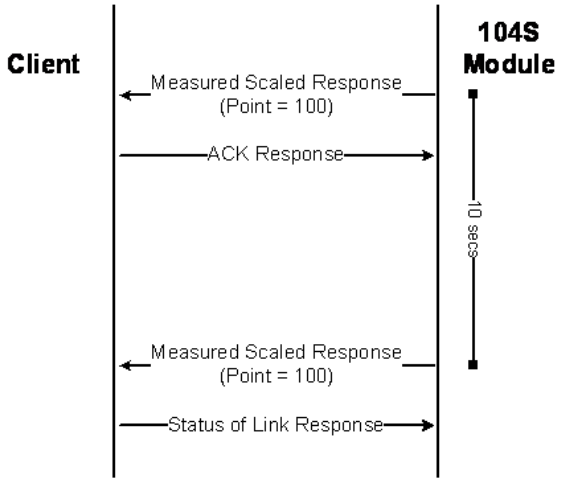


Example - Periodic Monitor Polling

The Cyclic Data Transmission parameter above is configured for 10 seconds (10000 milliseconds). If the following point is configured for monitor polling:



The following illustration shows the communication procedure:



Therefore, the point configured for a cyclic poll is periodically reported to the Master.

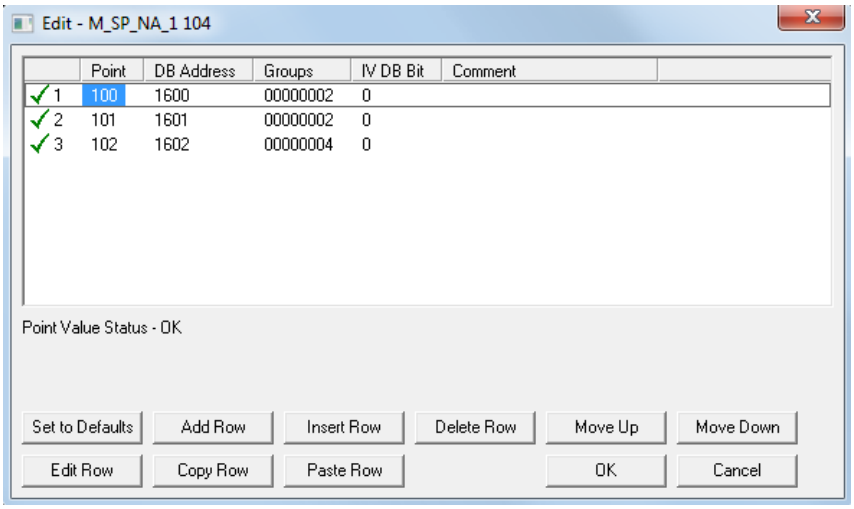
You may also create groups of points allowing the Master to poll certain points more frequently than other points. The Master may send requests for different groups as follows:

- General Interrogation (station)
- General Interrogation for Group 1
- General Interrogation for Group 2
- …
- General Interrogation for Group 16

Example - General Interrogation

If the following points are configured for General Interrogation:

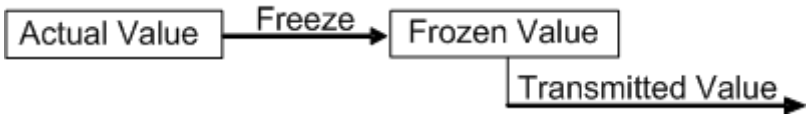
If you configure the following data points:



This feature allows you to separate the points into different groups according to the priority level that these should be reported to the Master. In the example above, points 100 and 101 would be returned with a General Interrogation for Group 1 and point 102 would be returned with a General Interrogation for Group 2.

Counter Points

There are four modes of acquisition of integrated totals (M_IT_NA points) defined by the protocol specification. The actual values may be memorized (copied) periodically to frozen values by a freeze command received from the master or initiated locally within the module.



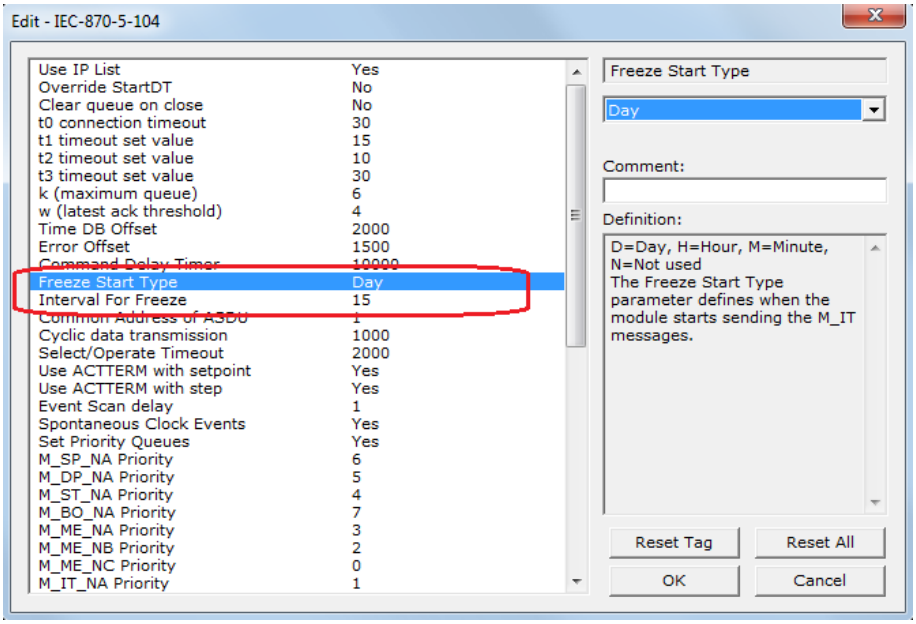
The gateway supports the following modes:

Mode A - Local freeze with spontaneous transmission

Mode D - Counter interrogation commands from the master initiate the freeze operation and the frozen values are reported spontaneously.

Example - Mode A

To use Mode A, configure the following parameters:



Freeze Start Type

The Freeze Start Type parameter will define when the gateway starts sending the M_IT messages.

Example I - Freeze Start Type

If the gateway powers up with the following date and time clock:

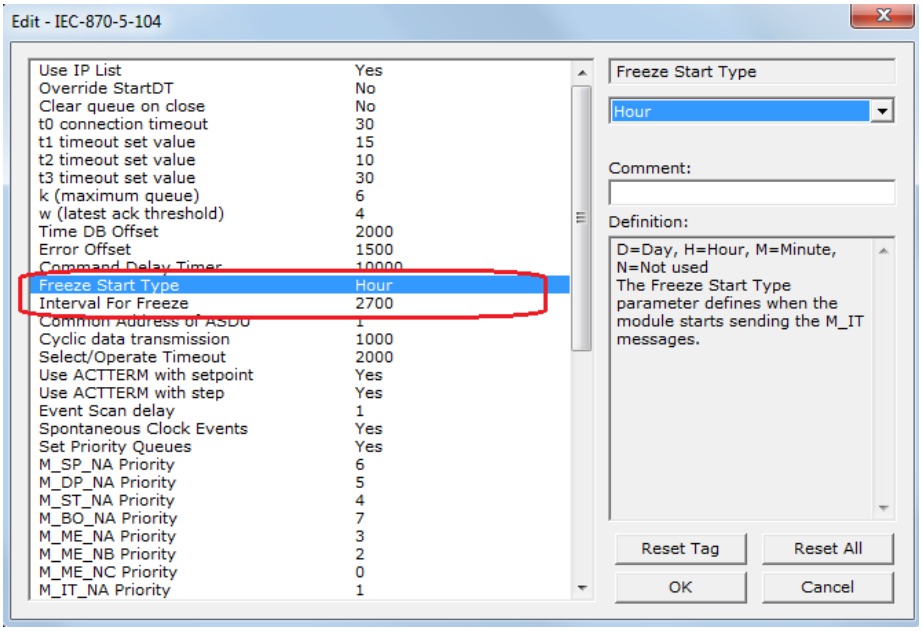
03/25/2004 18:07:42

If you configure the Interval For Freeze parameter at '15' (pictured above), the gateway would send the counter messages every 15 seconds. The gateway module gateway module freeze Start Type parameter as follows:

Freeze Start Type	Time to Start Sending Messages
D	03/26/2004 00:00:00
H	03/25/2004 19:00:00
M	03/25/2004 18:08:00

Example II - Freeze Start Type

If the gateway should send the counter points on the hourly turnaround time and also 45 minutes later, the Mode A parameters should be configured as follows:



So the gateway would send events as follows (Hours:Minutes:Seconds):

17:00:00
17:45:00
18:00:00
18:45:00
19:00:00
19:45:00

…

Mode D

To select the Mode D. configure the Freeze Start Type parameter as "N". For this mode the master would periodically send Counter Interrogation Commands to perform the freeze operation. After the values are frozen the gateway will return the counter points as events. The counter points must be properly configured for counter interrogation groups for Mode D operation.

Monitor Points Addressing

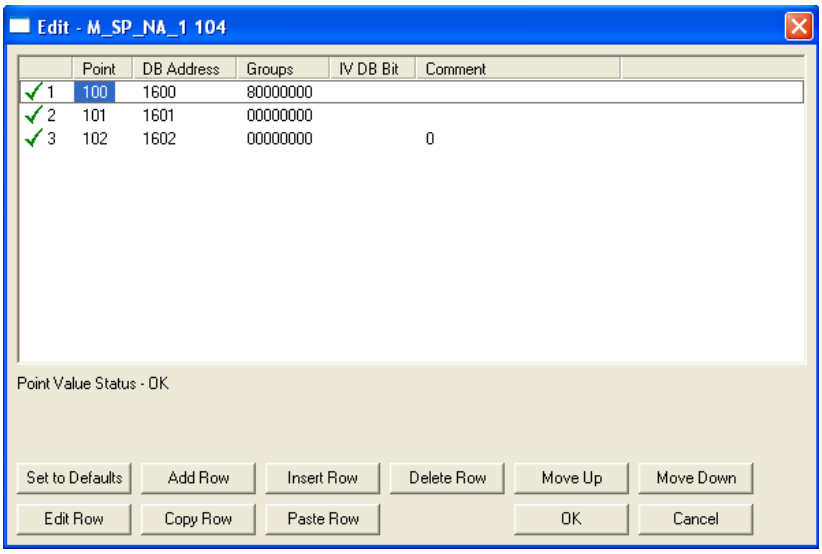
As discussed before, the monitor points must be configured in a database area in the PTQ-104S module.

The monitor data types are described in the following table.

Data Type	Data Size	Addressing Type
M_SP_NA	1 bit	Bit
M_DP_NA	2 bits	Bit
M_ST_NA	1 byte	Byte
M_ME_NA	1 word	Word
M_ME_NB	1 word	Word
M_ME_NC	2 word	Double-Word
M_IT_NA	2 word	Double-Word

M_SP_NA

A Monitored Single-Point occupies one binary bit and uses bit-addressing. For example, if you configured the following points as shown:



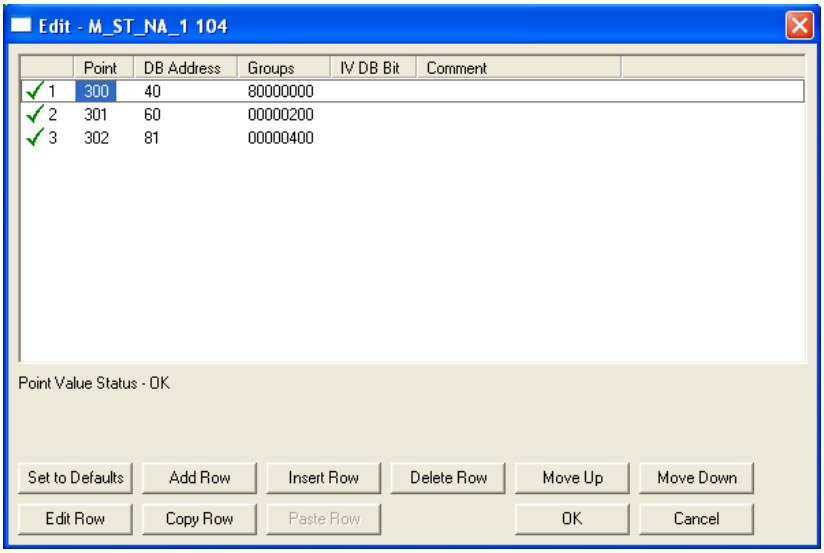
The following table describes how these points would be stored in the module:

100	Bit 0 of word 100 (Bit address 1600)
101	Bit 1 of word 100 (Bit address 1601)
102	Bit 2 of word 100 (Bit address 1602)

M_ST_NA

A Monitored Step-Point occupies one byte and uses byte-addressing.

For example, if you configured the following points:



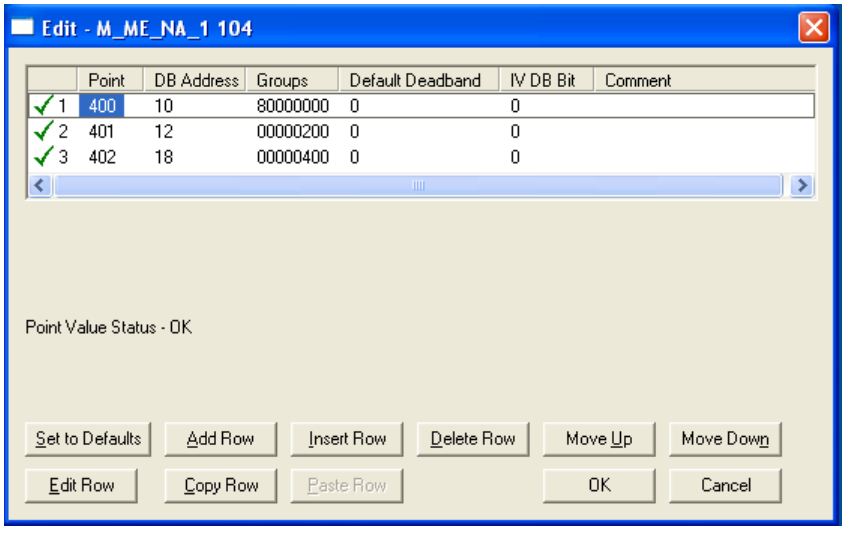
The following table describes how these points would be stored in the module:

300	Low Byte of word 20 (Byte address 40)
301	Low Byte of word 30 (Byte address 60)
302	High Byte of word 40 (Byte address 81)

M_ME_NA and M_ME_NB

A Monitored Measured Normalized point or Monitored Measured Scaled point occupies one word and uses word-addressing.

For example, if you configured the following points:



The following table describes how these points would be stored in the module:

400	Word 10 (Word address 10)
401	Word 12 (Word address 12)
402	Word 18 (Word address 18)

Monitored Measured Normalized points use a data representation defined by the protocol specification to represent fractional decimal values. The following table describes the value for each bit as a reciprocal power of two (2), that is two (2) raised to the power of a negative exponent (-1 through -15). Bit 15 is the Sign Bit.

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Value Hex(h) Decimal	Sign	4000h 2^{-1}	2000h 2^{-2}	1000h 2^{-3}	800h 2^{-4}	400h 2^{-5}	200h 2^{-6}	100h 2^{-7}	80h 2^{-8}	40h 2^{-9}	20h 2^{-10}	10h 2^{-11}	8h 2^{-12}	4h 2^{-13}	2h 2^{-14}	1h 2^{-15}
4000h 0.5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2000h 0.25	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1000h 0.125	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
6000h 0.75	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3210h 0.395751953125	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0

Examples:

A value of 4000hex (only Bit 14 set, all others clear) is interpreted as 0.5 decimal

A value of 2000hex (only Bit 13 set, all others clear) is interpreted as 0.25 decimal

A value of 1000hex (only Bit 12 set, all others clear) is interpreted as 0.125 decimal

... and so on until...

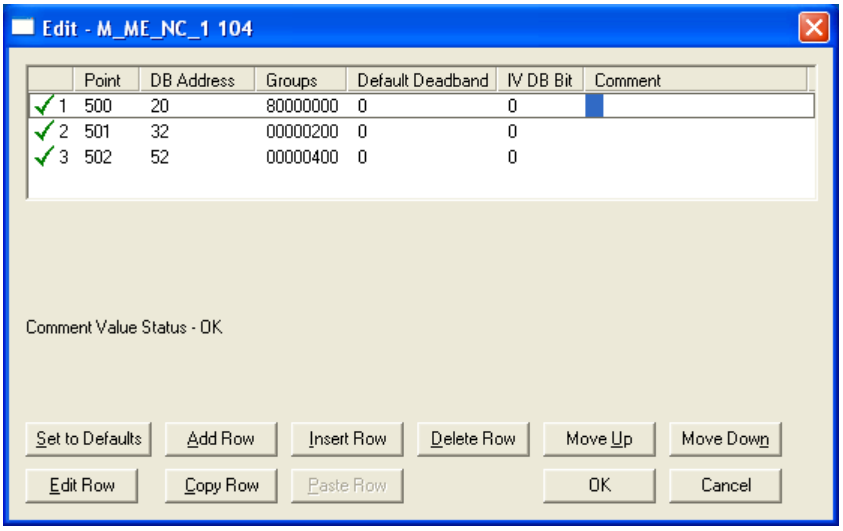
A value of 0001hex (Only Bit 0 set, all others clear) is interpreted as 0.000030517578125

Therefore, the actual data values transmitted may be any combination of the decimal values for any given bit pattern.

M_ME_NC and M_IT_NA

The monitored measured short floating point and monitored integrated total points occupy two words with double-word addressing.

For example, if you configured the following points:



The following table describes how these points would be stored in the module:

500	Words 40 and 41 (Double-word address 20)
501	Words 64 and 65 (Double-word address 32)
502	Word 104 and 105 (Double-word address 52)

9.2.4 Using Control (Command) Points

The following control points are supported by the PTQ-104S module:

C_DC_NA	Dual-Point Command
C_RC_NA	Step-Point Command
C_SE_NA	Measured Normalized Point Command
C_SE_NB	Measured Scaled-Point Command
C_SE_NC	Measured Short Floating-Point Command

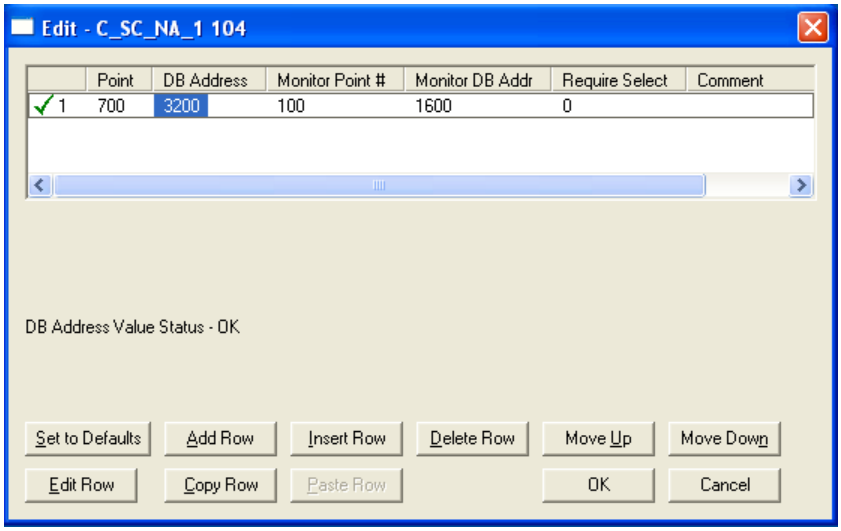
Each control point is identified by its Information Object Address. For each control point, configure the following parameters:

Point #: This is the information object address of the point. It identifies the point in the network. This address must be unique for each Common ASDU Address in the network.

DB Address: This is the database location in the PTQ-104S module associated with the Control Point.

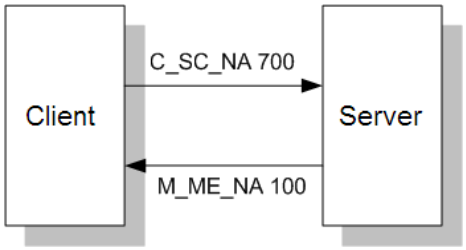
Monitor DB Address- The user might (optionally) configure a monitor point to be sent by the PTQ-104S module when it receives the command for that specific point.

Example (C_SC_NA)



In the example above, each time the module receives a command for single-command point 700, it sends a response containing a monitored single-point (information object address 100 with the value at database bit-address 1600).

Require Select: This parameter configures the point to require a *Select* request before the *Operate* command.



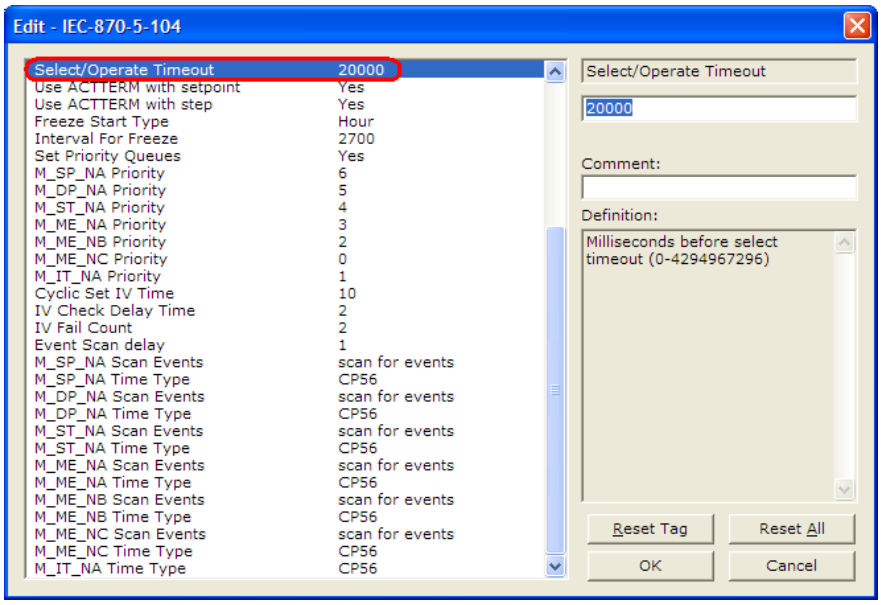
Control Data Transfer

The control communication typically occurs when the Client sends a command request to update the module's command points. The data types addressing are described in the following table.

Data Type	Data Size	Addressing Type
C_SC_NA	1 bit	Bit
C_DC_NA	2 bits	Bit
C_RC_NA	1 byte	Byte
C_SE_NA	1 word	Word
C_SE_NB	1 word	Word
C_SE_NC	2 words	Word

Some of the command points may be configured to be selected before executed.

Refer to the following parameter to configure the select/operate timeout period. After the module receives the SELECT operation it will wait for this period of time for the EXECUTE operation. If the module does not receive an EXECUTE operation within this period of time it will require another SELECT operation before the EXECUTE operation.



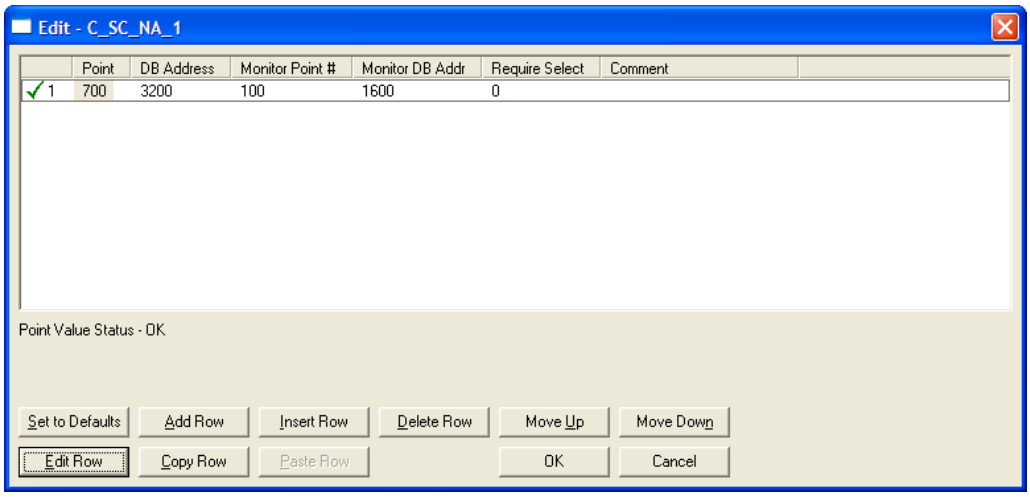
Command Points Addressing

Command points must be configured in a database area that is updated in the module. You must associate each point to a database address in the PTQ-104S module. The interpretation of the *DB address* parameter in the configuration tables depends on the point type configured and the type of addressing associated with that point type.

Data Type	Data Size	Addressing Type
C_SC_NA	1 bit	Bit
C_DC_NA	2 bits	Bit
C_RC_NA	1 byte	Byte
C_BO_NA	2 words	Double-word
C_SE_NA	1 word	Word
C_SE_NB	1 word	Word
C_SE_NC	2 words	Double-word

C_SC_NA and C_DC_NA

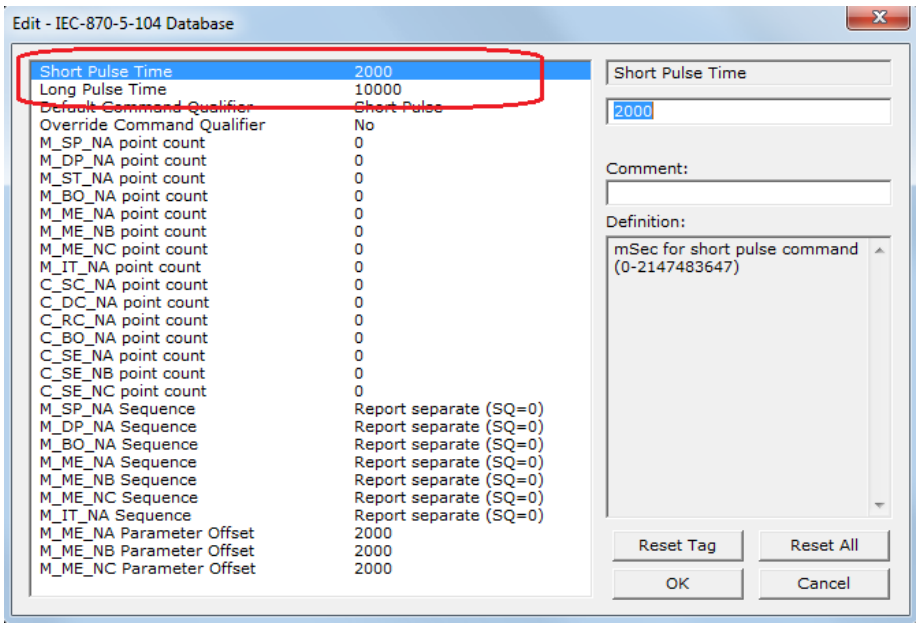
The single-point command and dual-point command points use one bit with bit addressing. For example, if you configure the following points:



These points would be used as follows:

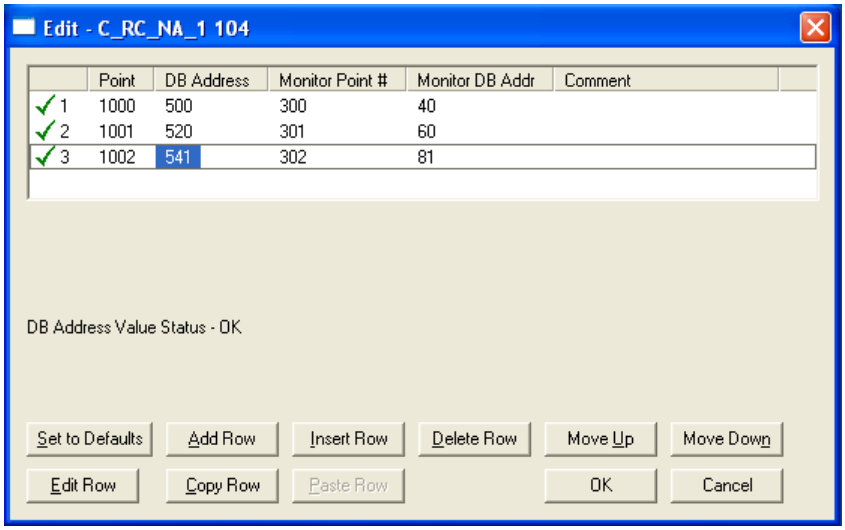
Inf. Object Address	Module Database Address
100	Bit 0 of word 100
101	Bit 1 of word 100
102	Bit 2 of word 100

The protocol specification defines a qualifier value that is set by the Master to determine the duration of the pulse (short, long or persistent). Configure the parameters below to set the duration of the short and long pulses:



C_RC_NA

A Step-Point Command point occupies one byte and uses byte addressing. For example, if you configured the following points:

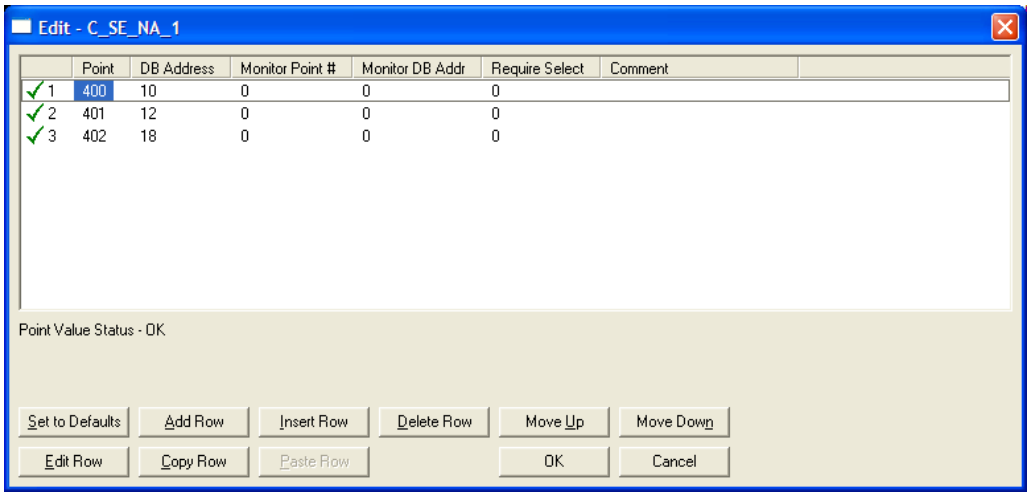


The following table describes how these points would be used.

Inf. Object Address	Module Database Address
1000	Low Byte of word 250 (Byte address 500)
300	Low Byte of word 20 (Byte address 40. Must match configuration of point 300 in M_ST_NA.)
1001	Low Byte of word 260 (Byte address 520)
301	Low Byte of word 30 (Byte address 60. Must match configuration of point 301 in M_ST_NA.)
1002	High Byte of word 270 (Byte address 541)
302	High Byte of word 40 (Byte address 81. Must match configuration of point 302 in M_ST_NA.)

C_SE_NA and C_SE_NB

The measured normalized point command uses one word with word-addressing. For example, if you configured the following points:



The following table describes how these points would be used.

Inf. Object Address	Module Database Address
1100	Word 2000 (Word address 2000)
400	Word 10 (Word address 10. Must match the configuration of point 400 in M_ME_NA.)
1101	Word 2001 (Word address 2001)
401	Word 12 (Word address 12. Must match the configuration of point 401 in M_ME_NA.)
1102	Word 2001 (Word address 2001)
402	Word 18 (Word address 18. Must match the configuration of point 402 in M_ME_NA.)

The measured normalized points use a data representation defined by the protocol specification to represent fractional decimal values. The following table describes the value for each bit as a reciprocal power of two (2), that is two (2) raised to the power of a negative exponent (-1 through -15). Bit 15 is the Sign Bit.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value Hex(h) Decimal	Sign	4000h 2^{-1}	2000h 2^{-2}	1000h 2^{-3}	800 h 2^{-4}	400 h 2^{-5}	200 h 2^{-6}	100 h 2^{-7}	80 h 2^{-8}	40 h 2^{-9}	20 h 2^{-10}	10 h 2^{-11}	8h 2^{-12}	4h 2^{-13}	2h 2^{-14}	1h 2^{-15}
4000h 0.5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2000h 0.25	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1000h 0.125	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
6000h 0.75	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
3210h 0.395751953125	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0

Examples:

A value of 4000hex (only Bit 14 set, all others clear) is interpreted as 0.5 decimal

A value of 2000hex (only Bit 13 set, all others clear) is interpreted as 0.25 decimal

A value of 1000hex (only Bit 12 set, all others clear) is interpreted as 0.125 decimal

... and so on until...

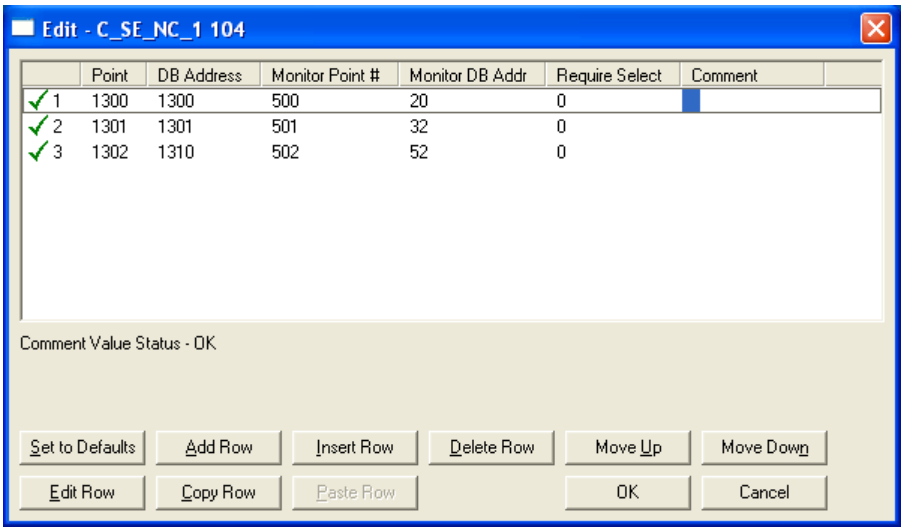
A value of 0001hex (Only Bit 0 set, all others clear) is interpreted as 0.000030517578125

Therefore, the actual data values transmitted may be any combination of the decimal values for any given bit pattern.

C_SE_NC

A Measured Short Floating- point Command point occupies two words and uses double word addressing.

For example, if you configured the following points:



The following table describes how these points would be stored in the module:

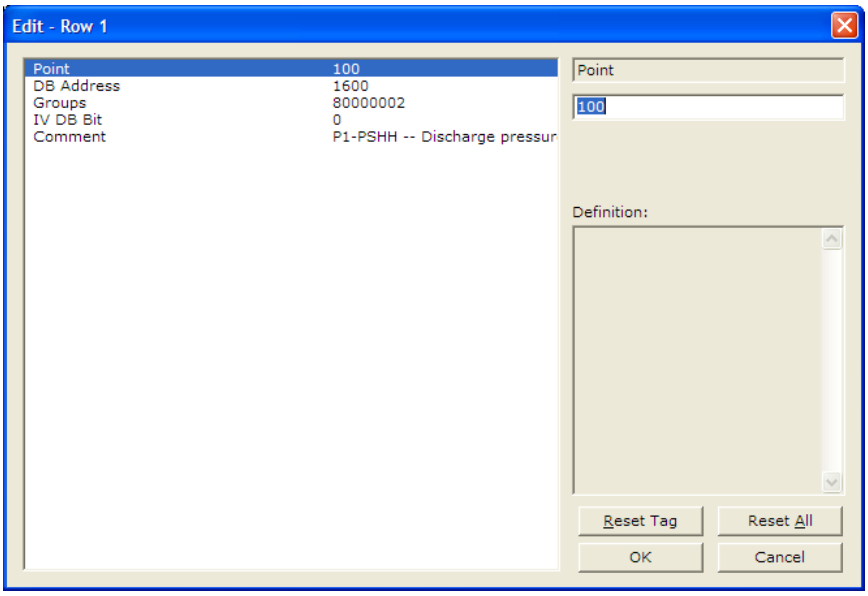
1300	Words 2600 and 2601 (Double-word address 1300)
500	Words 40 and 41 (Double-word address 20. Must match the configuration of point 500 in M_ME_NC.)
1301	Words 2602 and 2603 (Double-word address 1301)
501	Words 64 and 65 (Double-word address 32. Must match the configuration of point 501 in M_ME_NC.)
1302	Words 2620 and 2621 (Double-word address 1310)
502	Word 104 and 105 (Double-word address 52.Must match the configuration of point 502 in M_ME_NC.)

9.2.5 Data Communication

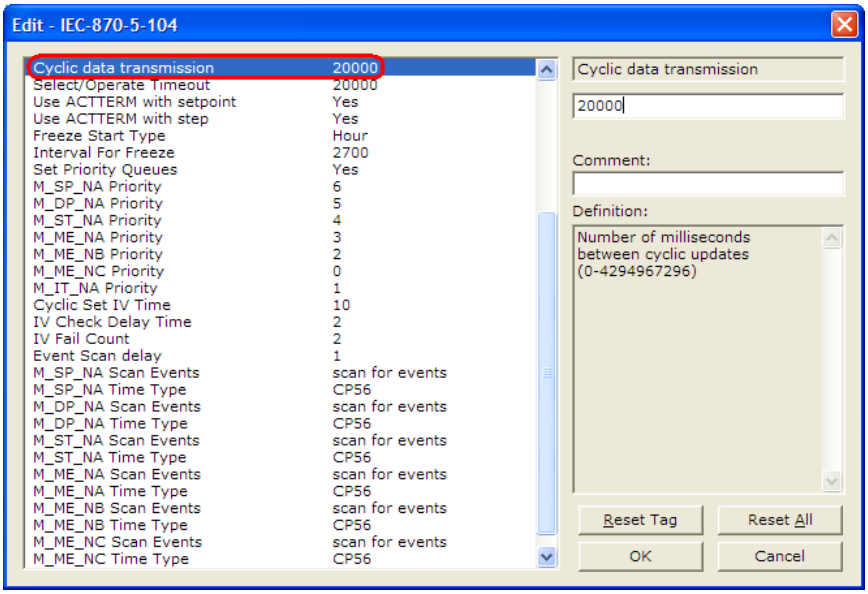
Group Communication

The Group parameter in the module configuration file controls how each monitored point is transferred between the PTQ-104S and the client unit. The Group parameter is described in detail in Group Definition (page 92)

The following example configures this point to be repeated either during cyclic polls, or when the module General Interrogation request for group 1 occurs.



The module periodically sends all points configured for periodic/cyclic poll (0x80000000) at the interval in milliseconds configured with the following parameter:

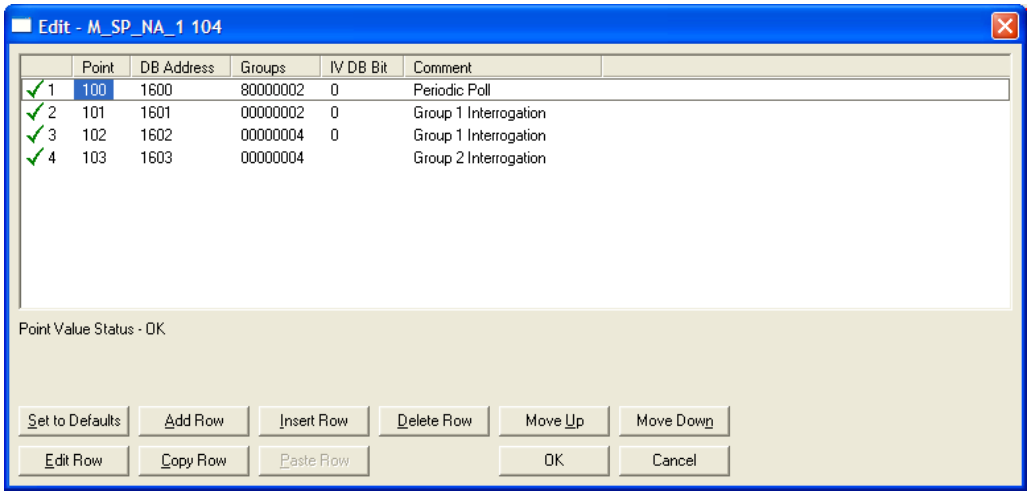


You can also divide the monitored points into different groups, allowing the client to periodically poll only certain points. This also allows some points to be polled more frequently than others.

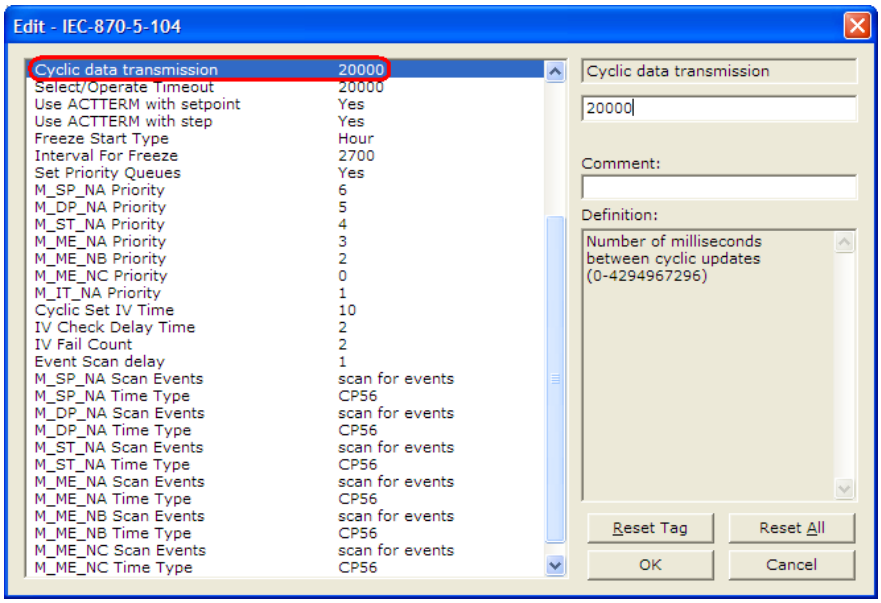
Note: You should configure the counter points (M_IT_NA) for general counter interrogation or group counter interrogations.

Example:

In the following example, with the following data points for data type M_SP_NA configured:



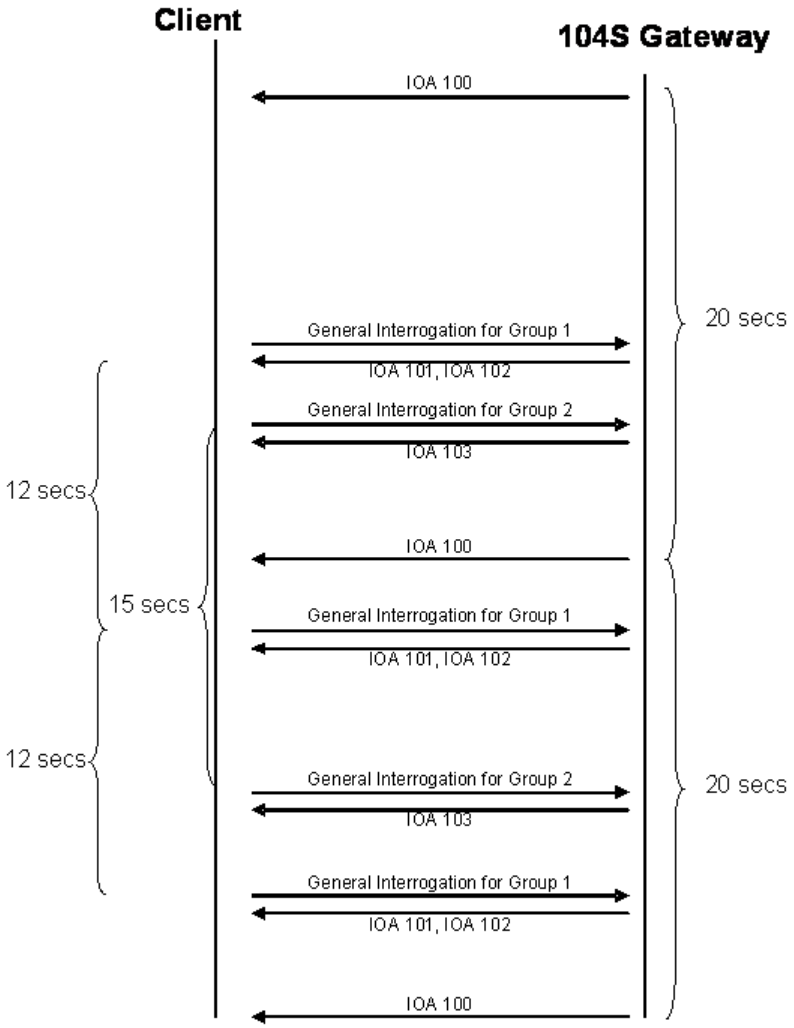
And the Cyclic data transmission parameter configured as follows:



The client unit sends the following requests:

- General Interrogation for Group 1 every 12 seconds
- General Interrogation for Group 2 every 15 seconds

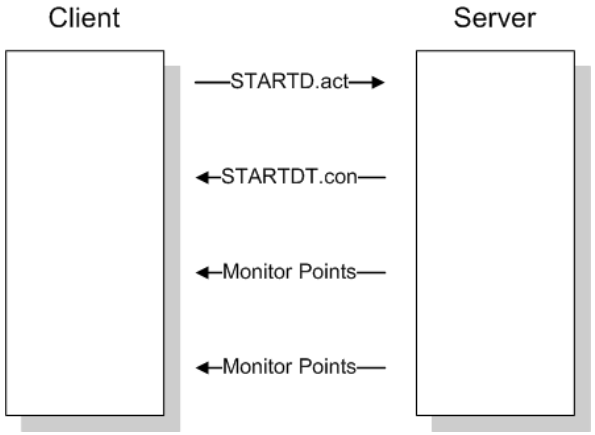
The following illustration shows how the communication would be performed between the client and the gateway.



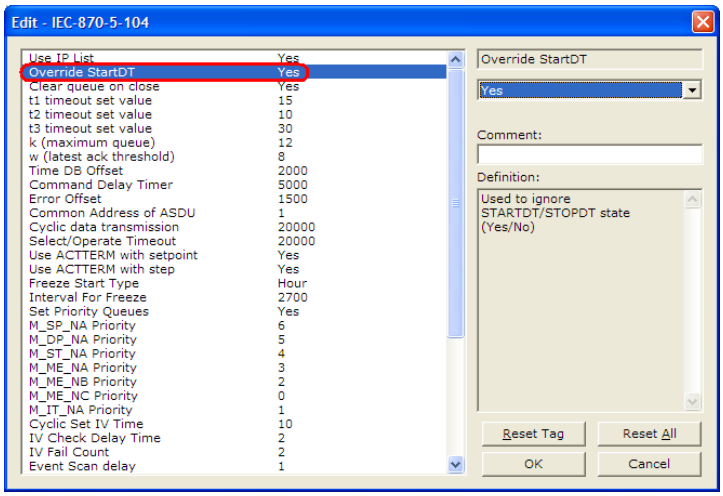
STARTDT & STOPDT

STARTDT (Start Data Transfer) and STOPDT (Stop Data Transfer) are used by the client to control the data transfer from the PTQ-104S module. When the connection is established, user data is not automatically enabled in the server until it receives a STARTDT act request from the client. The server should respond with a STARTDT con response to acknowledge the Client request.

Once this procedure is concluded, the server can send monitor data to the client. The Client can interrupt the monitor data flow at any time by sending a STOPDT act command to the server.



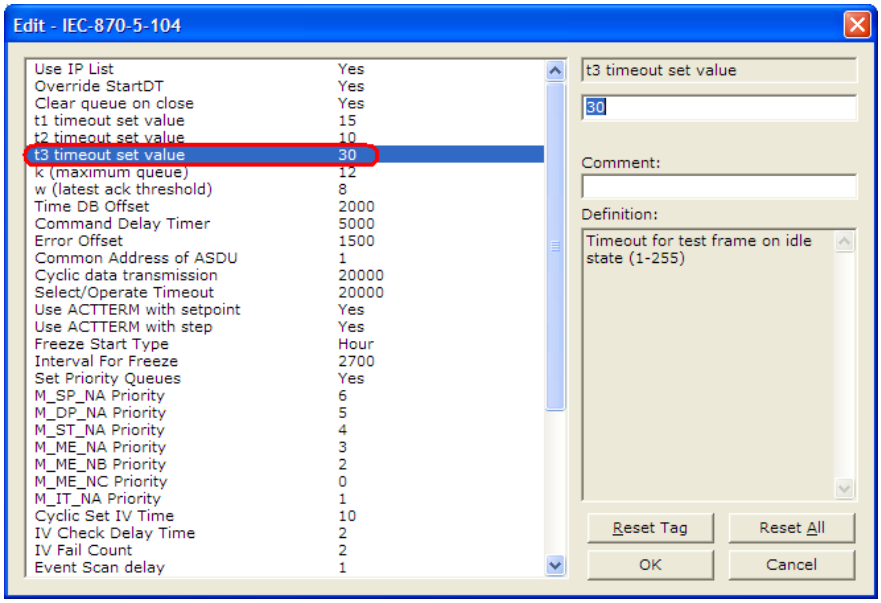
In some circumstances the Client unit may not support STARTDT and STOPDT messages. The module may also be tested with simulator software that does not support these features. During these situations, you may want to disable the STARTDT and STOPDT features using the following parameter:



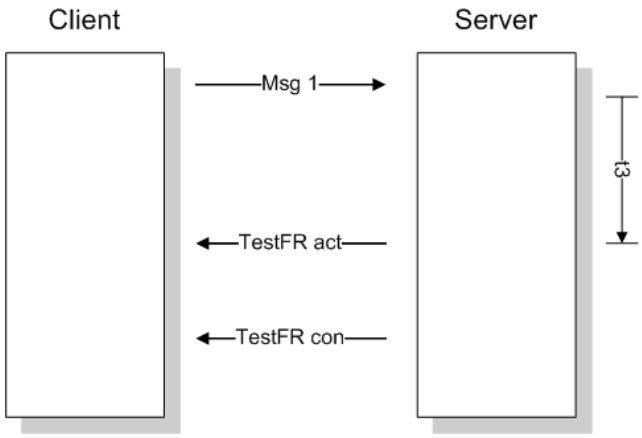
If this parameter is set to **YES (1)**, the module will ignore the STARTDT and STOPDT requests by the Client unit.

TESTFR Requests

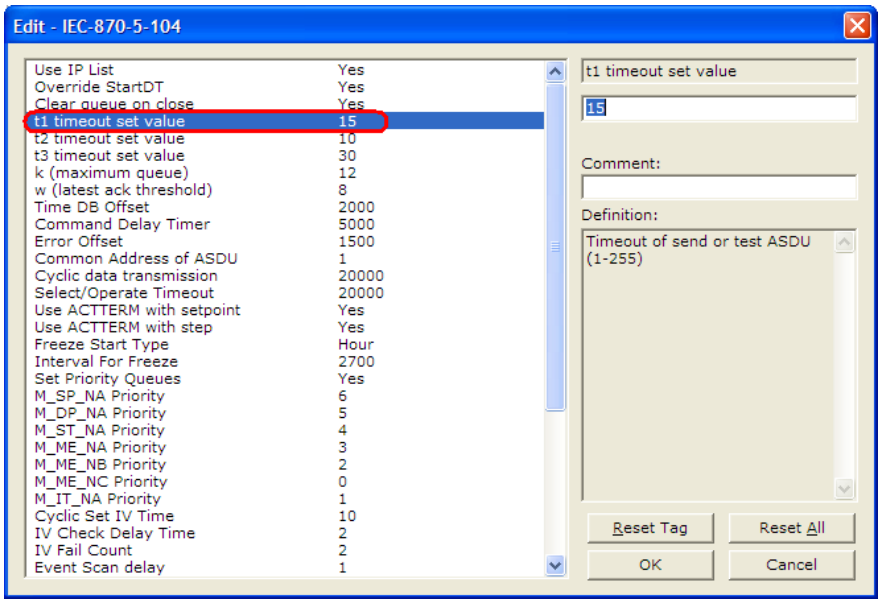
Connections that are unused (but opened) may be periodically tested in both directions by sending test messages (TESTFR=act) which are confirmed by the receiving station sending TESTFR=con messages. The PTQ-104S module can be configured to periodically send this message using the following parameter:

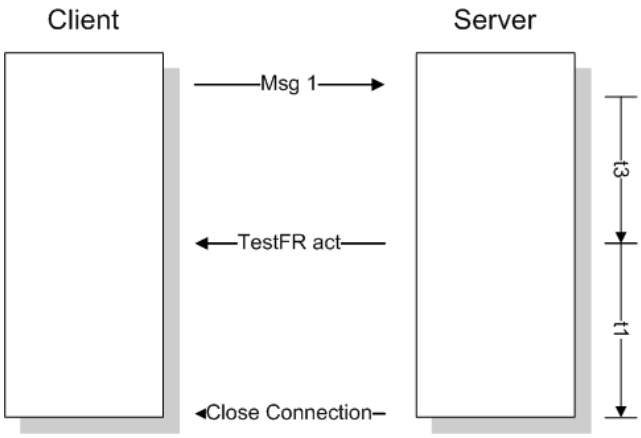


In the example above, the module would send a TESTFR.ACT message 30 seconds after receiving the last message:

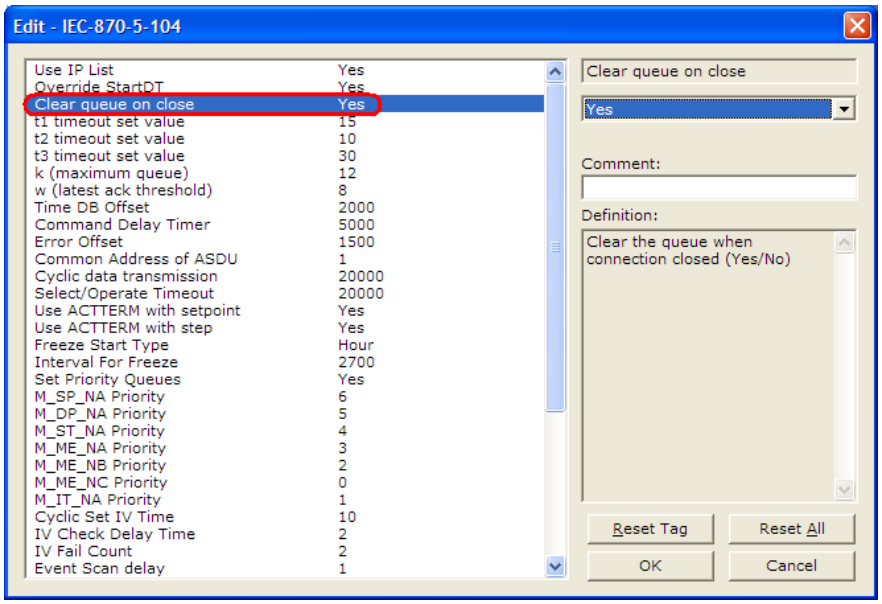


If the module does not receive the TESTFR.con message within a certain amount of time, it will timeout and close the connection. You can configure the timeout period using the following parameter:





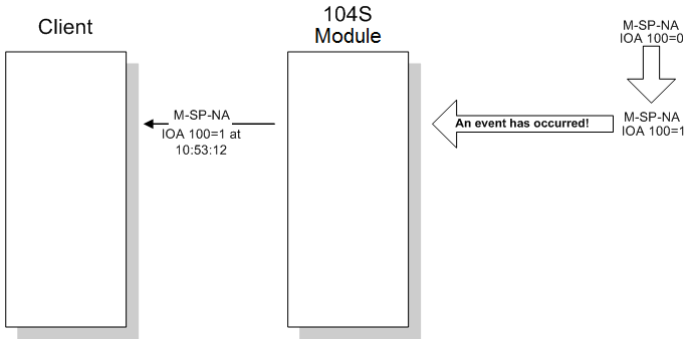
When closing the connection, the module can be configured to clear all the messages in its queue. The following parameter is used to implement this task:



The above configuration would cause the module to delete all pending messages/events while closing the connection to the Client.

9.2.6 Events

In order to improve communication efficiency, most applications will require the Client to periodically poll for data changes with a higher priority than polling for other monitored data. Every time a data point changes, the server can send this information as an *event*, typically with date and time information indicating when the change occurred.

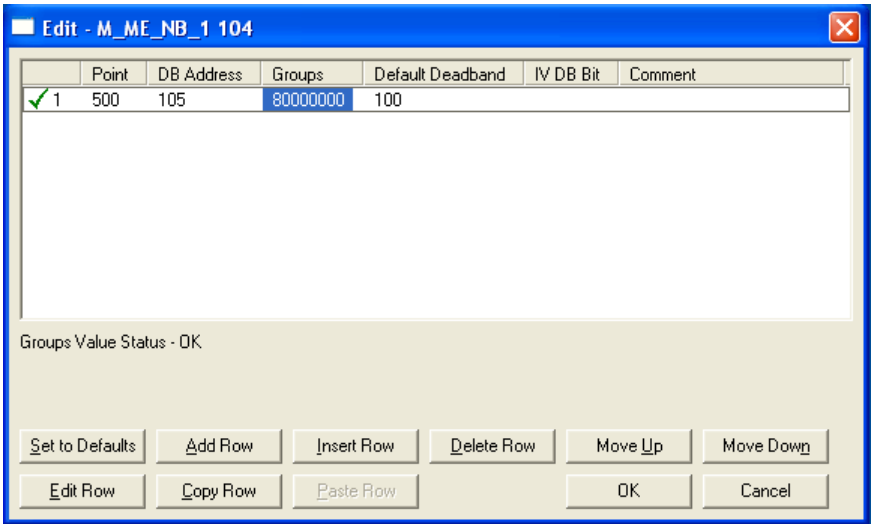


The module supports a buffer queue of 99 events per data type. When the queue is full, the module will delete the oldest event in the queue, replacing it with the newest event received for that data type.

Deadbands

The Monitored Measured points (M_ME_NA and M_ME_NB) will generate events only if the data value changes an amount greater than or equal to the configured deadband value.

For example, with the following point configured:



If the current value for this point is 130, it would only generate an event, only if:

NEW VALUE is less than or equal to 30

OR

NEW VALUE is greater than or equal to 230.

You can set the deadband for each Monitored Measured point through the configuration file.

The Client may also dynamically change the deadband for each Monitored point. The Client may send one of the following commands:

Type	Command
110	Parameter of Measured Normalized Data (M_ME_NA)
111	Parameter of Measured Scaled Data (M_ME_NB)
112	Parameter of Measured Short Floating Point (M_ME_NC)

The protocol specification explains that the qualifier value for these commands should be configured as:

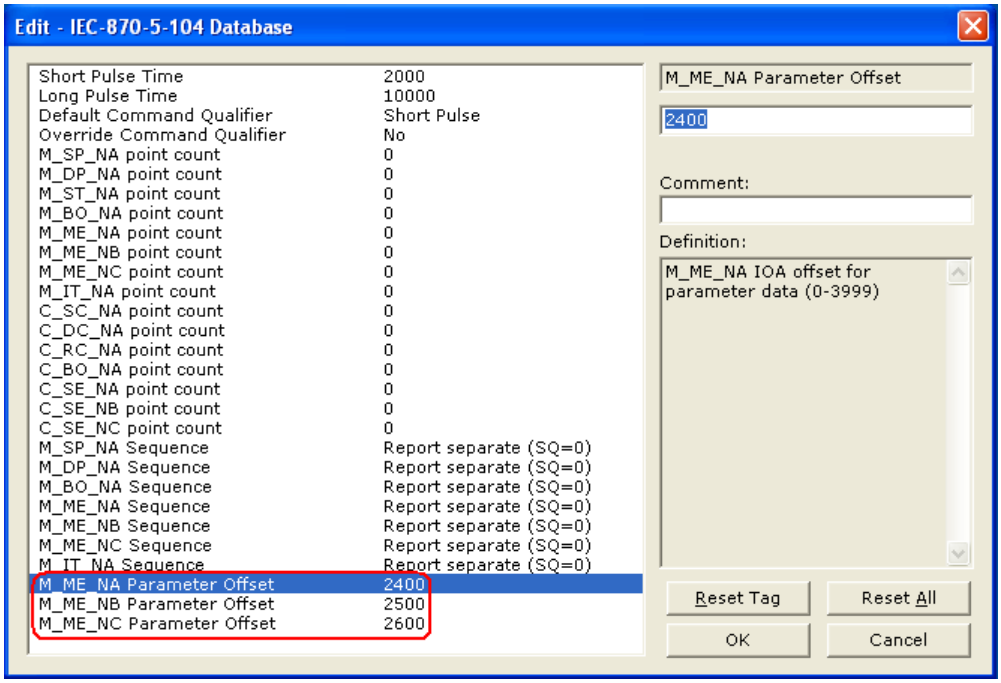
Bits	Value	Description
1 to 6	0	Not Used
	1	**Threshold Value (Deadband)**
	2	Smoothing Factor (filter time constant) - Not Supported
	3	Low Limit Transmission of Measured Value
	4	High Limit Transmission of Measured Value
	5..31	Reserved
7	0	No Change
	1	Change
8	0	Operation
	1	Not in Operation

For the PTQ-104S gateway, the Low Limit and High Limit parameters cannot be changed by command, because these values are calculated as follows:

Low Limit = (LAST REPORTED VALUE) - Deadband

High Limit = (LAST REPORTED VALUE) + Deadband

These commands must be sent to a specific Information Object Address. The PTQ-104S gateway associates each Monitor Measured point with a Parameter point using *Parameter Offset* entries:



Example:

Assume the following Monitored Measured points are configured as shown:

```
[M_ME_NA_1 104]
#                                  Default
# IOA       DB Address   Group(s)  Deadband  IV DB Bit
# -------   ----------   --------  --------  ---------
START
      400           10   00000002       100        # P1 suction pressure
      401           11   00000002       100        # P1 discharge pressure
      402           12   00000002       100        # P2 suction pressure
      403           13   00000002       100        # P2 discharge pressure
      404           14   00000002       100        # Station discharge
pressure
      405           15   00000002       100        # VSD speed
      406           16   00000002       100        #
      407           17   00000002       100        #
      408           18   00000002       100        #
      409           19   00000002       100        #
[M_ME_NB_1 104]
#
#                                   Default
# IOA       DB Address   Group(s)  Deadband  IV DB Bit
# -------   ----------   --------  --------  ---------
START
      500           20   00000002       100       # P1 inboard bearing temp
      501           21   00000002       100       # P1 outboard bearing temp
      502           22   00000002       100       # P1 winding Temp
      503           23   00000002       100       # P1 current
      504           24   00000002       100       # P2 inboard bearing temp
      505           25   00000002       100       # P2 outboard bearing temp
      506           26   00000002       100       # P2 winding Temp
      507           27   00000002       100       # P2 current
      508           28   00000002       100       #
      509           29   00000002       100       #
[M_ME_NC_1 104]
#                                   Default
# IOA       DB Address   Group(s)  Deadband  IV DB Bit
# -------   ----------   --------  --------  ---------
START
      600           30   00000002       100             #
      601           32   00000002       100             #
      602           34   00000002       100             #
      603           36   00000002       100             #
      604           38   00000002       100             #
      605           40   00000002       100             #
      606           42   00000002       100             #
      607           44   00000002       100             #
      608           46   00000002       100             #
      609           48   00000002       100             #
```

And assume the Parameter Offsets are configured as shown:

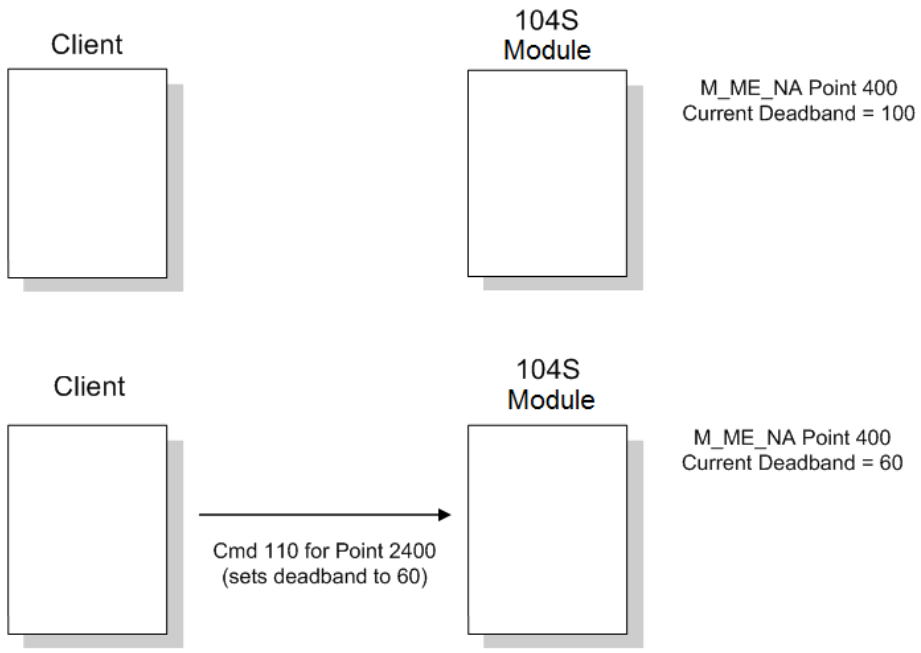
```
M_ME_NA Parameter Offset : 2400 #M_ME_NA IOA offset for parameter data
M_ME_NB Parameter Offset : 2500 #M_ME_NB IOA offset for parameter data
M_ME_NC Parameter Offset : 2600 #M_ME_NC IOA offset for parameter data
```

It would imply that the parameter points would be configured as follows:

M_ME_NA

Monitored Measured Normalized Point	Associated Parameter Point
400	2400
401	2401
402	2402
403	2403
...	...
409	2409

In order to send a change of deadband for M_ME_NA Point 400, the Client would send a command type 110 to point 2400.



M_ME_NB

Monitored Measured Normalized Point	Associated Parameter Point
500	2500
501	2501
502	2502
503	2503
...	...
509	2509

In order to send a change of deadband for M_ME_NB Point 500, the Client would send a command type 110 to point 2500.

M_ME_NC

Monitored Measured Normalized Point	Associated Parameter Point
600	2600
601	2601
602	2602
603	2603
...	...
609	2609

In order to send a change of deadband for M_ME_NC Point 600, the Client would send a command type 110 to point 2600.

Controlling the Generation of Events

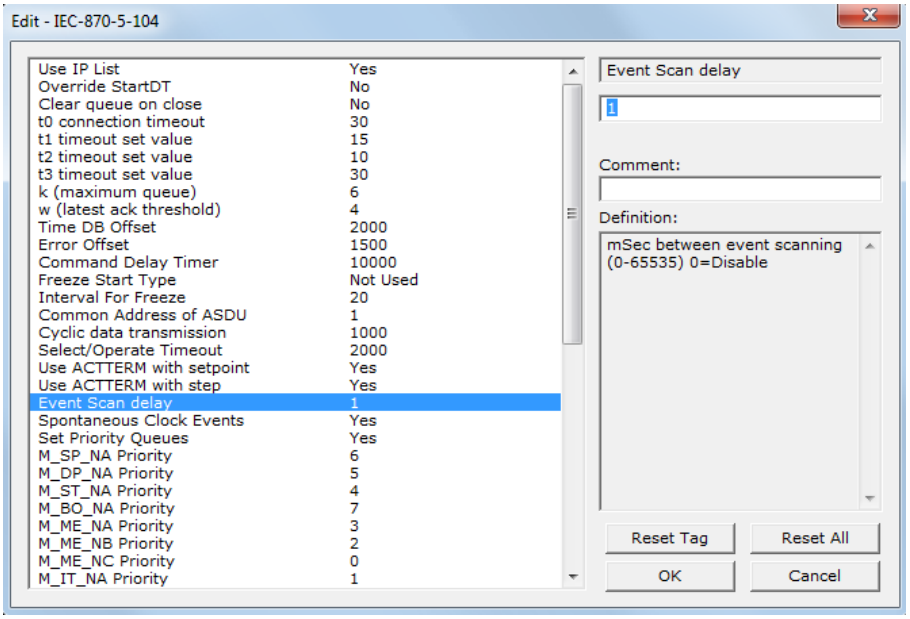
Some applications may require that only some points should generate events. The application will receive current values from points which have not been configured to generate events only by issuing a poll request, or what the protocol specification calls an *interrogation*. Other applications may require that all configured points generate events.

The PTQ-104S module offers much flexibility for event control. You may control whether or not events will be generated at three (3) different levels:

1 At the General Application Level (all configured points of all types)
2 At the Data Type Level (all configured points of any data type)
3 At the Individual Point Level (any specific point of any data type)

General (All Points)

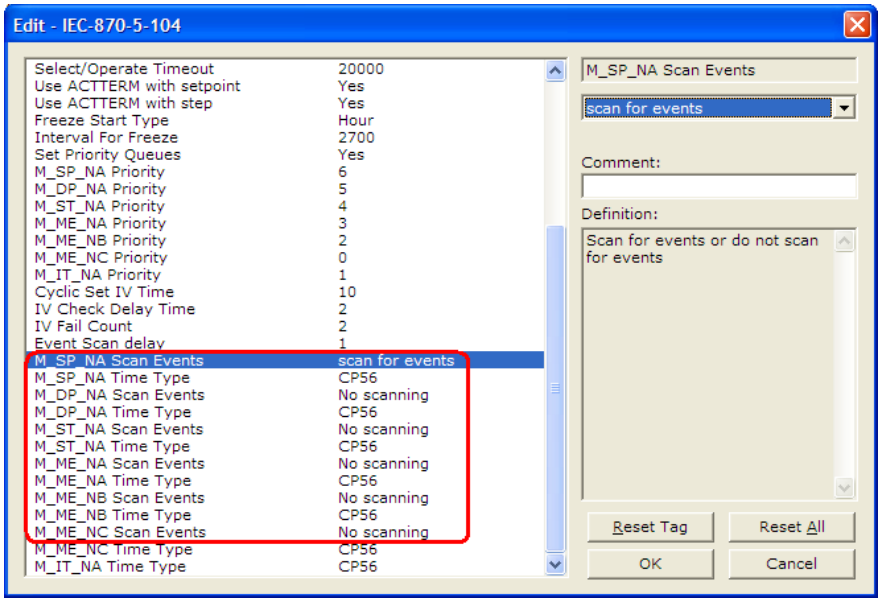
The user may control how frequently the module will scan the database for events using the following configuration parameter:



If this parameter is set to 0, the module will not generate events for any points. A non-zero value will configure how frequently the module will scan for events in the database.

Data Type Level

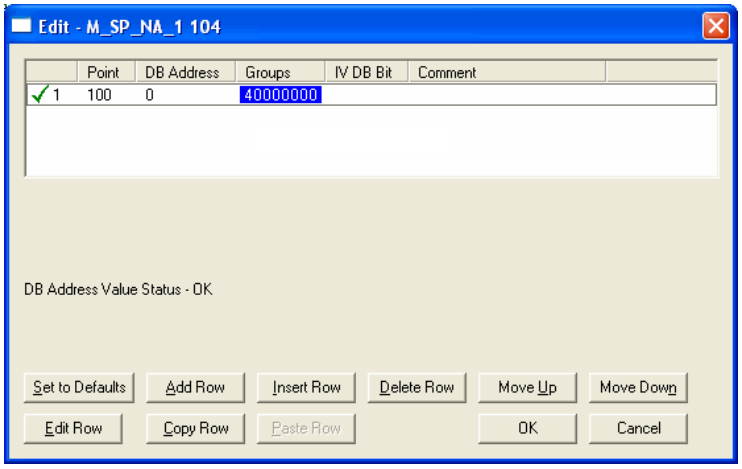
The user may configure if a data type should generate events or not. Each data type has a configuration parameter to control the generation of events:



In the example above, only the M_SP_NA points would generate events.

Point Level

You can configure whether or not each point should generate events on a point-by-point basis by using the Group field for each point configuration. Set the value as 40000000 to disable the generation of events for any specific point.

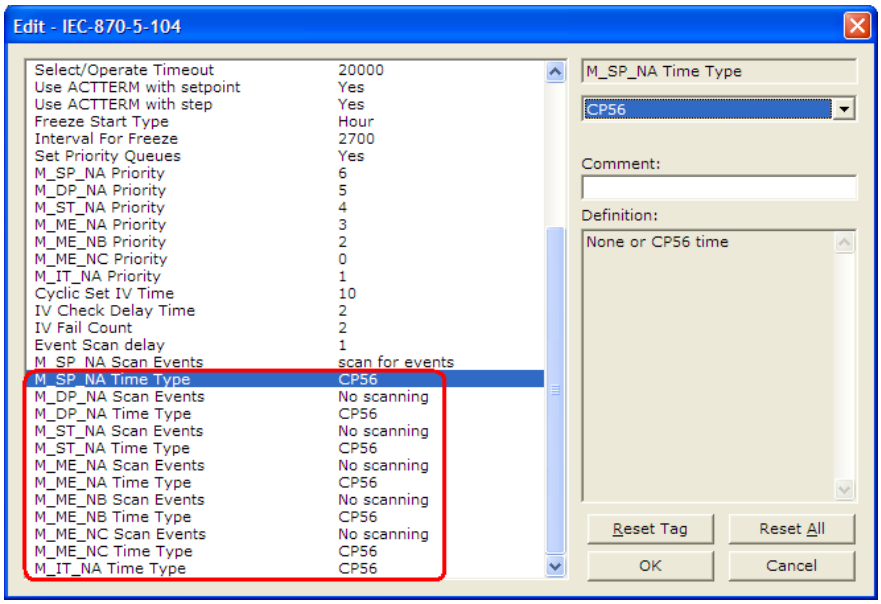


Time Information

Each event may also send the date and time when it occurred. The PTQ-104S module supports the CP56 time format (as defined in the protocol specification). This format contains the millisecond, second, minute, hour, day, month and year when the event occurred.

The PTQ-104S gateway may also be configured not to send any time information with each event for certain data types.

The following "Time Type" parameters may be used to control the time information for each data type:



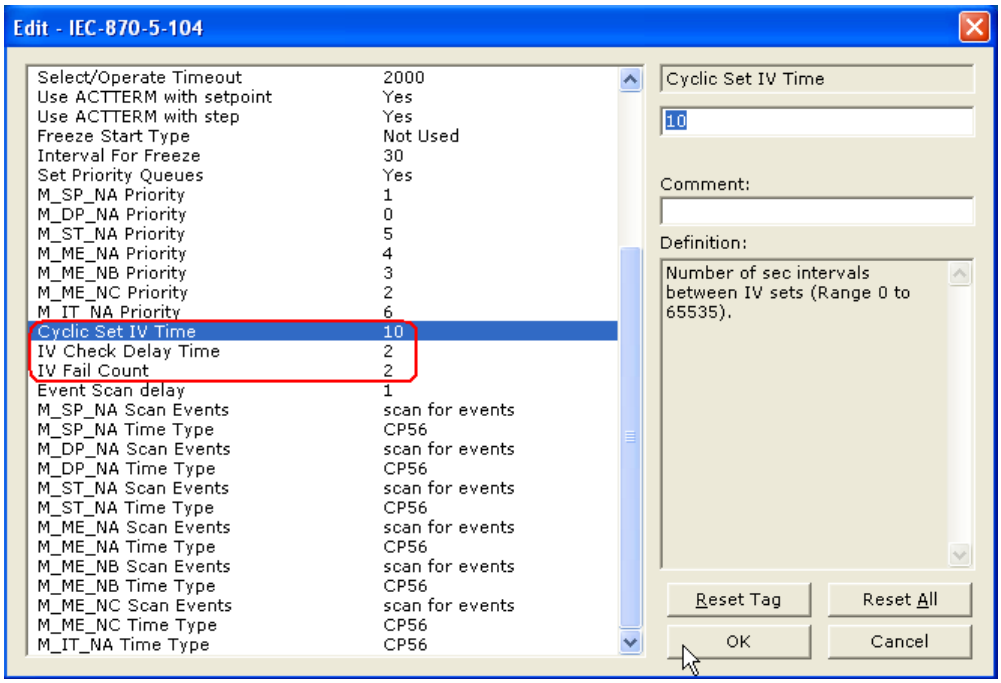
Note: The Client should send a Time Synchronization command to the module in order to synchronize its date and time information, according to the protocol specifications. Depending on certain parameters, as well as hardware limitations, the module may present some time delay over time. The Client should periodically send time synchronization requests to the 104S module.

Invalid Bit Monitoring

This feature allows the application to set the invalid (IV) quality bit of the protocol for all the monitored ASDU types supported. If the feature is enabled, the status of the IV quality bit for each point configured by the user can be individually monitored and controlled.

The parameters required to support this feature are:

- Cyclic Set IV Time
- IV Check Delay Time
- IV Fail Count

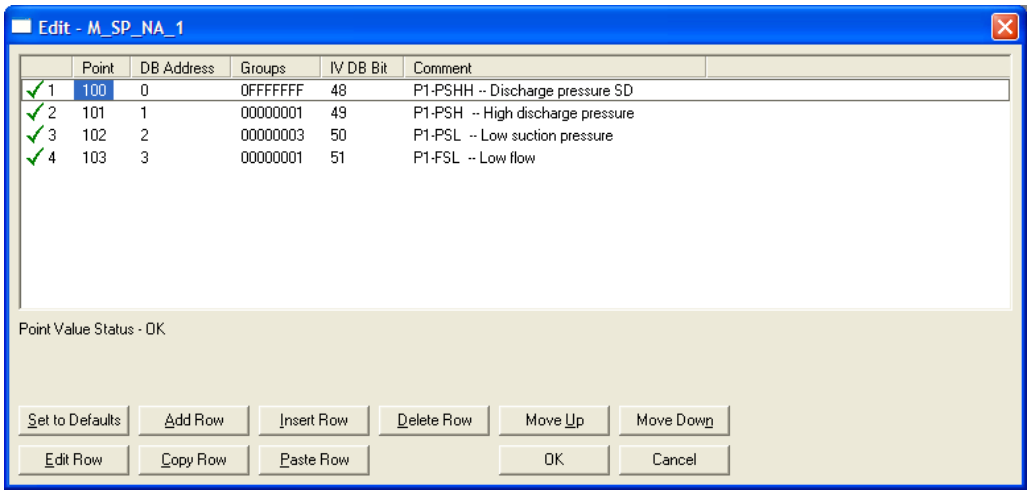


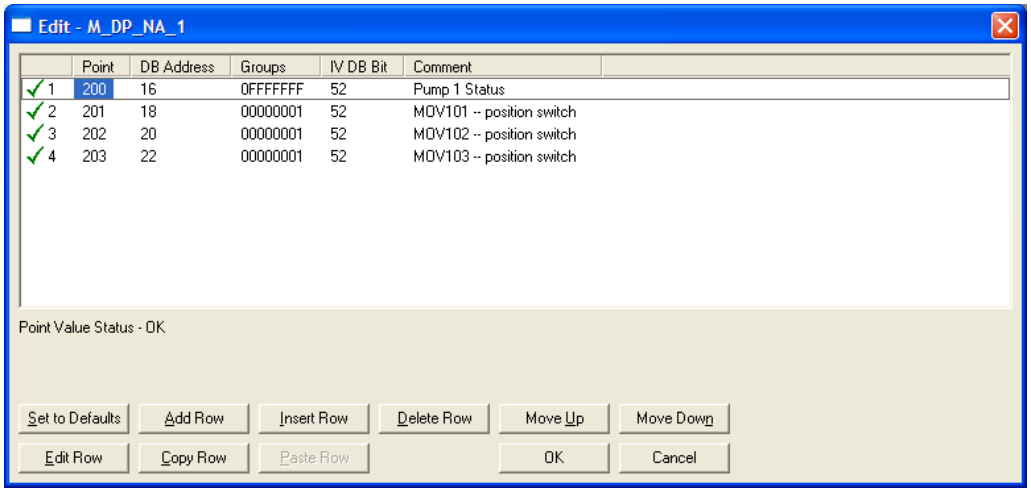
To disable this feature, set the IV Fail Count parameter to 0. If the IV bit field is absent or set to 0, the invalid quality state for the point will always be reported as valid. If used, the Cyclic Set IV Time parameter must be at least 3 times larger than the IV Check Delay Time.

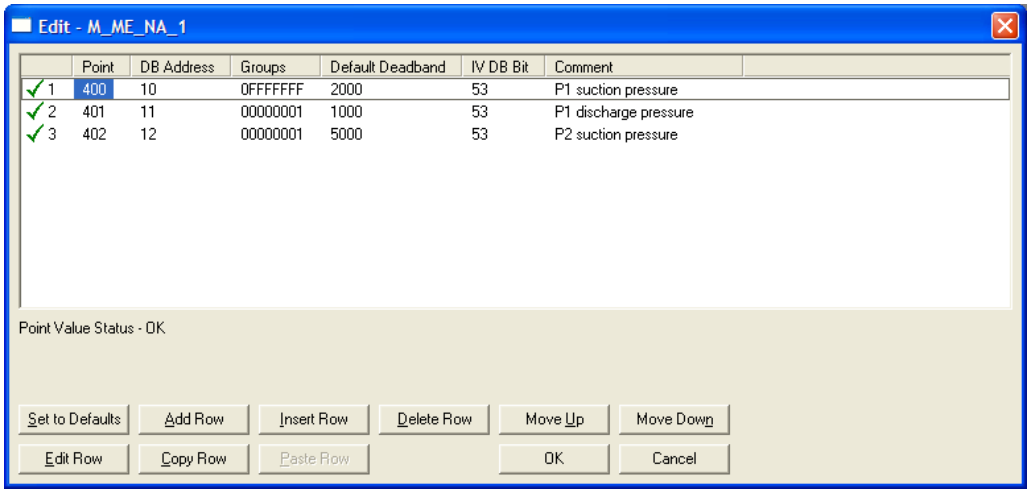
The **CYCLIC SET IV TIME** parameter determines the interval at which the driver will set all the IV bits for the points being monitored. If the IV bit is ON for the number of times specified by the **IV FAIL COUNT** parameter, the point is in an invalid state. The driver will check the state of each bit at the frequency determined by the **IV CHECK DELAY TIME**.

The values for the parameters must permit the driver to properly execute the logic. For example, the value for **CYCLIC SET IV TIME** should be twice the **IV CHECK DELAY TIME,** multiplied by the **IV FAIL COUNT**. If the cyclic timer is set to a smaller value, the logic may not execute correctly.

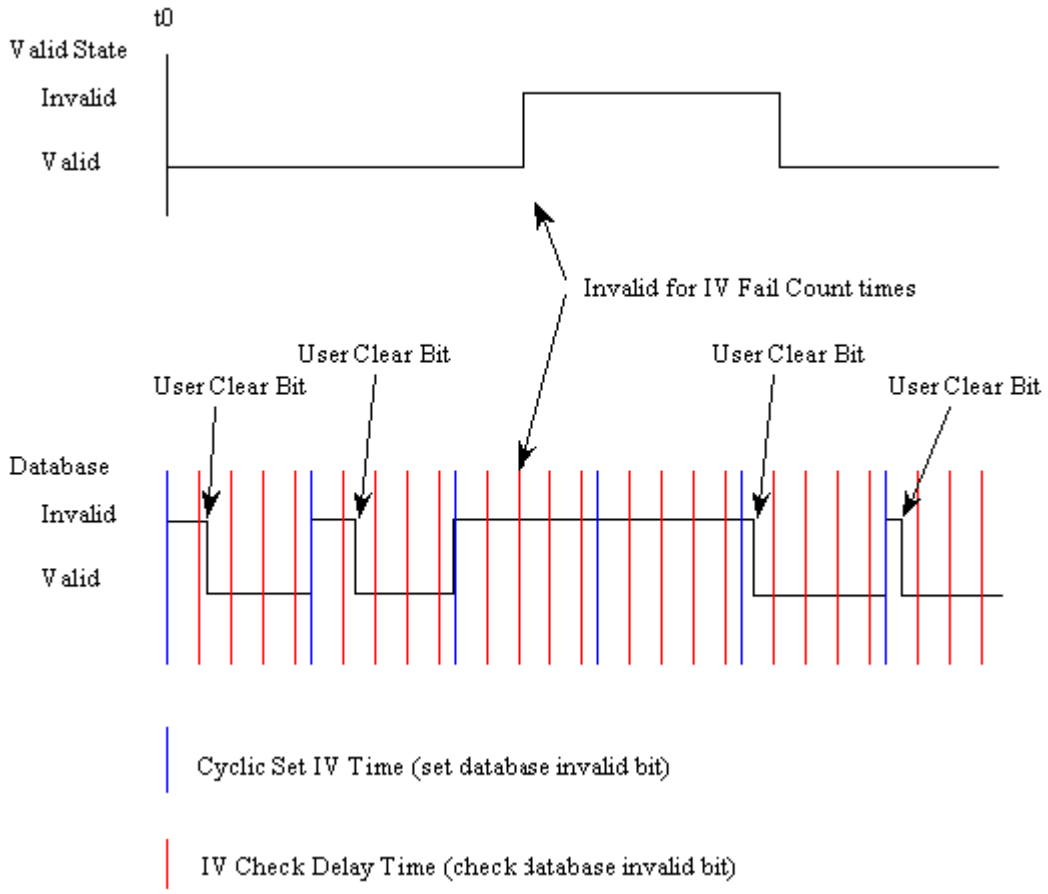
In order to configure points for this feature, the **IV DB BIT** field for each data type must be set to the database configuration records in the configuration file. For example:







The following illustration shows how these parameters are implemented:

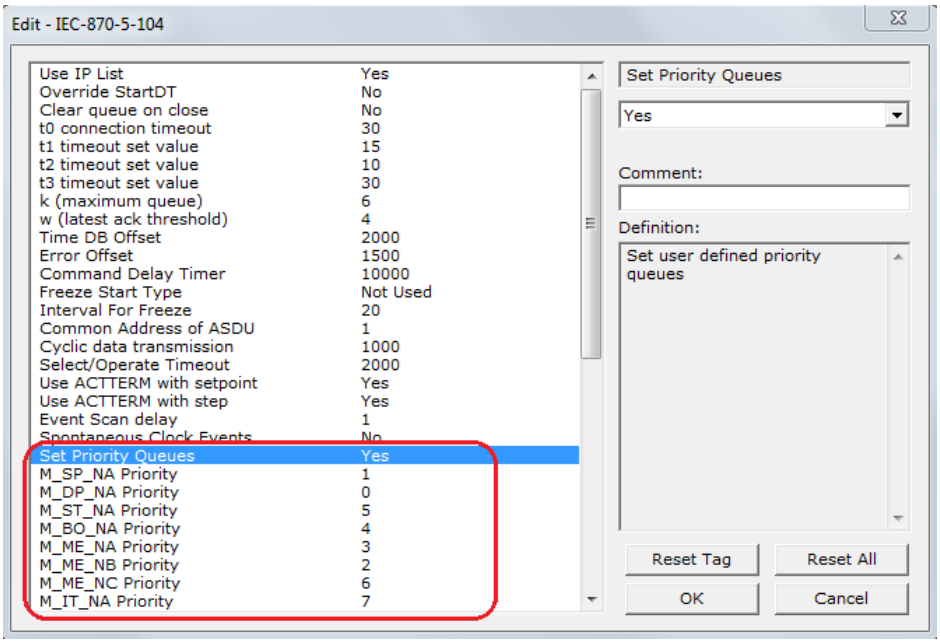


If a database bit address (1 to 64000) is present, the application may consider the point with an invalid flag if the previous logic checks the IV bit as 1 during consecutive IV Check Delay scans. The IV bits would have to be reset to 0 to set the point to valid state.

The IV DB bit defined for each point can be unique, or many points may share the same bit. The last case could be used when the all points on an I/O module are to be considered as one set. In this case only a single bit is required. For a point that is the result of a computation, the valid quality state could be set for each point individually.

Event Priority

Event Priority permits ASDUs that generate events to be placed in priority queues that are set by the user. The configuration file contains the following parameters to support this feature:



The Set Priority Queues parameter must be enabled for this feature to be used. Each of the ASDUs affected by this feature must be assigned a unique priority index from 0 to 7. When events of the ASDU with a priority of 0 are present, they will always be reported before events from any other ASDUs with higher numbers.

Example - Event Priority

Using the event priority values in the example above, if the event queue (requested by the processor and sent to the module) contains the events in the following order:

Event Order	ASDU
1	M_SP_NA
2	M_SP_NA
3	M_DP_NA
4	M_ST_NA
5	M_DP_NA
6	M_SP_NA

The module will respond to a class one data request from the controlling station by returning the data in the event queue in the order shown in the following table.

Packet Order	Content
1	M_DP_NA events 3 and 5
2	M_SP_NA events 1, 2, and 6
3	M_ST_NA event 4

The module responds in order of the priority settings in the configuration, not the order in which it was requested from the processor.

Note that the events are packed into messages in order to maximize the efficiency of the network. The following warning must be considered when deciding to use this feature: Because events from the highest priority queues are always reported when present before lower priority queues, events in the lower queues may be lost due to buffer overflow.

If this feature is not utilized, each of ASDU's events is stored in its own queue. The module will report each queue containing events by reporting events from each queue, one queue after the other, in a repeating cycle with all the data for each ASDU being packed into one or more messages. This methodology limits the possibility of a buffer overflowing and still maximizes the use of bandwidth on the communication channel.

SNTP Support

SNTP is used for time synchronization of produced and consumed commands. When an exchange occurs the driver compares time stamps from the previous exchange. When the new exchange time is less than the previous exchange, the exchange is ignored. This can occur when the Ethernet packets are routed and delayed. Time synchronization provides for data integrity. The following table lists the parameters defined in this section:

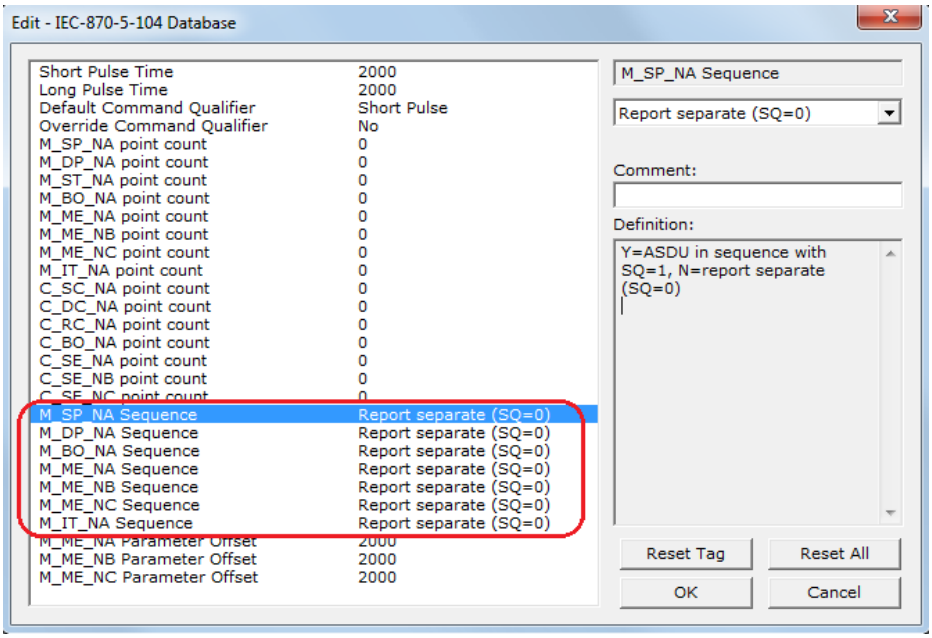
The SNTP driver will compute a new clock value every 5 minutes using the average value of 10 samples each collected over an approximate 6-second period. This new value will be used to adjust the clock maintained by the SNTP driver and used by the application. If a valid database register is specified, the driver will place the time value into the module’s database. The first two registers will contain the number of seconds and the next two registers will contain the number of microseconds since January 1, 1970.

A list of some of the common NTP servers can be obtained at http://www.ntp.org/, http://www.eecis.udel.edu/~mills/ntp/servers.html, along with the appropriate IP address. Other server lists can be found on the Internet by searching on "NTP Servers" with your browser.

9.2.7 Sequence Flag

YES or **NO**

In order to save bandwidth, you can configure the Sequence Flag feature using the following parameters:



If this parameter is set to **NO**, this feature is not selected, the module will send the object address and value for every monitored point sent to the Client, and the Sequence Flag (SQ) will be set to zero (0).

If this parameter is set to **YES**, this feature is selected, the module will send the object point address and value for only the first point of a sequence of points, send only the data value without point address for any remaining points in the sequence, and the Sequence Flag (SQ) will be set to one (1). When SQ = 1, the Client assumes that all points after the first point use information object addresses in a contiguous order (using the first point as the reference starting address).

Note: Refer to the Client device specification to verify if this feature is supported before you consider using it.

9.3 Cable Connections

The PTQ-104S module has the following communication connections on the module:

- One Ethernet port (RJ45 connector)
- One RS-232 Configuration/Debug port (DB9 connector)

9.3.1 Ethernet Connection

The PTQ-104S module has an RJ45 port located on the front of the module, labeled *Ethernet*, for use with the TCP/IP network. The module is connected to the Ethernet network using an Ethernet cable between the module's Ethernet port and an Ethernet switch or hub.

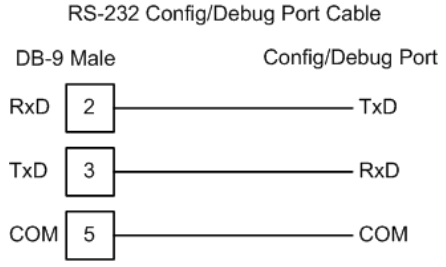
Note: Depending on hardware configuration, you may see more than one RJ45 port on the module. The Ethernet port is labeled *Ethernet*.
Warning: The PTQ-104S module is NOT compatible with Power Over Ethernet (IEEE802.3af / IEEE802.3at) networks. Do NOT connect the module to Ethernet devices, hubs, switches or networks that supply AC or DC power over the Ethernet cable. Failure to observe this precaution may result in damage to hardware, or injury to personnel.
Important: The module requires a static (fixed) IP address that is not shared with any other device on the Ethernet network. Obtain a list of suitable IP addresses from your network administrator BEFORE configuring the Ethernet port on this module.

Ethernet Port Configuration - wattcp.cfg

The wattcp.cfg file must be set up properly in order to use a TCP/IP network connection (page 18). You can view the current network configuration in the PCB Diagnostics window (page 120).

9.3.2 RS-232 Configuration/Debug Port

This port is physically an RJ45 connection. An RJ45 to DB-9 adapter cable is included with the module. This port permits a PC based terminal emulation program to view configuration and status data in the module and to control the module. The cable for communications on this port is shown in the following diagram:



9.4 Error Status Table

This section contains a listing of the PTQ-104S module's status data area, copied automatically to the database starting at address 4000. To request the complete status block, refer to Read Status (9250) on page 64.

Word Offset in Block	Data Field(s)	Description
4000	Scan Count	This status value contains a counter incremented on each scan of the module's main loop.
4001 to 4002	Product Name	This two-word data area contains the text values representing the product name.
4003 to 4004	Revision	This two-word data area contains the text values for the revision number.
4005 to 4006	Op Sys #	This two-word data area contains the text values for the operating system number.
4007 to 4008	Run Number	This two-word data area contains the text values for the run number.
4009	Reserved	Reserved
4010	Block Reads	
4011	Block Writes	
4012	Blocks Parsed	
4013	Block Errors	
4020	Send Sequence	This word contains the current Send Sequence number (NS) sent by the module. This value is a part of every message sent to the client unit.
4021	Receive Sequence	This word contains the current Receive Sequence number (NR) sent by the module. This value is a part of every message sent to the client unit.
4022	Ack Sequence	This value is incremented every time the messages sent by the module are acknowledged. The client acknowledges a number of APDUs when it returns the Receive Sequence Number up to the number whose APDUs are properly received. The module will hold the APDUs in a buffer until it receives back its own Send Sequence Number as a Receive Sequence Number. The buffer status information is available at the following status parameters: (In Queue, Queue First, Queue Index)
4023	Received Packets	This word contains the number of messages received by the client but without being acknowledged by the module
4024	In Queue	This word contains the number of messages in the queue that have not been acknowledged by the client yet.
4025	Queue First	This word should contain a value of 0.
4026	Queue Index	This word contains the current index number of the latest message in the buffer queue.
4027	Tx Count	Number of messages received
4028	Rx Count	Number of messages sent
4029	t0 Timeout	Number of t0 errors recognized by the module
4030	t1 Timeout	Number of t1 errors recognized by the module
4031	t2 Timeout	Number of t2 errors recognized by the module
4032	t3 Timeout	Number of t3 errors recognized by the module

Word Offset in Block	Data Field(s)	Description
4033	Sequence Number Errors	This word contains the number of sequence number errors recognized by the module. When the send sequence number received by the module does not match the expected sequence number, the connection is closed and this counter is incremented
4034	Bad Address	This word contains the number of messages received with a wrong Common ASDU Address value
4035	Bad Length	This word contains the number of messages received by the host that does not contain a valid length field
4036	Active	!0 = Yes, 0 = No.
4037	Socket State	This word contains the current socket state as follows: -1 = Open Socket 0 = Wait for Connection 1 = Transmit Message if Ready 2 = Receive Packet and process message 3 = Process multiple messages in packet 50 = Send TestFR ACT 51 = Wait for TestFR Con 60 = Send S-Format Message 1000 = Close Socket 1001 = Wait for Socket to Close
4038	Socket Open Count	This word contains the number of times the socket listen function was executed
4039	Socket Close Count	This word contains the number of times the an active close function was executed
4040	Socket Connect Count	This word contains the number of times a connection was established between with the remote host unit
4041	Event Buffer 0	Event data from MSP free events
4042	Event Buffer 1	Event data from MDP free events
4043	Event Buffer 2	Event data from MST free events
4044	Event Buffer 3	Event data from MMA free events
4045	Event Buffer 4	Event data from MMB free events
4046	Reserved	Reserved
4047	Event Buffer 6	Event data from MIT free events

The module has two parameters to copy parts of the status registers to the database as follows:

[Backplane Configuration] section: Error Offset

Word Offset in Block	Data Field(s)	Description
0	Scan Count	This status value contains a counter incremented on each scan of the module's main loop.
1 to 2	Product Name	This two-word data area contains the text values representing the product name.
3 to 4	Revision	This two-word data area contains the text values for the revision number.

Word Offset in Block	Data Field(s)	Description
5 to 6	Op Sys #	This two-word data area contains the text values for the operating system number.
7 to 8	Run Number	This two-word data area contains the text values for the run number.
9	Reserved	Reserved

[IEC-870-5-104] section: Error Offset

Word Offset in Block	Data Field(s)	Description
0	t0 Timeout	Number of t0 errors recognized by the module
1	t1 Timeout	Number of t1 errors recognized by the module
2	t2 Timeout	Number of t2 errors recognized by the module
3	t3 Timeout	Number of t3 errors recognized by the module
4	Sequence Number Errors	This word contains the number of sequence number errors recognized by the module. When the send sequence number received by the module does not match the expected sequence number, the connection is closed and this counter is incremented
5	Bad Address	This word contains the number of messages received with a wrong Common ASDU Address value
6	Bad Length	This word contains the number of messages received by the host that does not contain a valid length field
7	Active	!0 = Yes, 0 = No.
8	Socket State	This word contains the current socket state as follows: -1 = Open Socket 0 = Wait for Connection 1 = Transmit Message if Ready 2 = Receive Packet and process message 3 = Process multiple messages in packet 50 = Send TestFR ACT 51 = Wait for TestFR Con 60 = Send S-Format Message 1000 = Close Socket 1001 = Wait for Socket to Close
9	Socket Open Count	This word contains the number of times the socket listen function was executed
10	Socket Close Count	This word contains the number of times the an active close function was executed
11	Socket Connect Count	This word contains the number of times a connection was established between with the remote host unit

9.5 Group Codes

One aspect of the point configuration database that leads to confusion is the group definition field, which assigns each point to one or more interrogation groups. Use of interrogation groups permits the controlling unit to interface with a specific set of data. Refer to the IEC 60870-5-104 standard for a full discussion of interrogation groups. A specific group, Periodic data group, reports data points on a set frequency. The frequency is set in the **Cyclic Data Transmission** parameter in the configuration file. Remember that a point can be assigned to more than one group.

Group Code	Description
0x00000001	Interrogated by general interrogation (station or global)
0x00000002	Interrogated by group 1 interrogation
0x00000004	Interrogated by group 2 interrogation
0x00000008	Interrogated by group 3 interrogation (See **Note 1**, below)
0x00000010	Interrogated by group 4 interrogation
0x00000020	Interrogated by group 5 interrogation
0x00000040	Interrogated by group 6 interrogation
0x00000080	Interrogated by group 7 interrogation
0x00000100	Interrogated by group 8 interrogation
0x00000200	Interrogated by group 9 interrogation
0x00000400	Interrogated by group 10 interrogation
0x00000800	Interrogated by group 11 interrogation
0x00001000	Interrogated by group 12 interrogation
0x00002000	Interrogated by group 13 interrogation
0x00004000	Interrogated by group 14 interrogation
0x00008000	Interrogated by group 15 interrogation
0x00010000	Interrogated by group 16 interrogation
0x00020000	Interrogated by general counter request
0x00040000	Interrogated by group 1 counter request
0x00080000	Interrogated by group 2 counter request
0x00100000	Interrogated by group 3 counter request
0x00200000	Interrogated by group 4 counter request
0x40000000	Disable event scanning of this point
0x80000000	Periodic/cyclic data returned from unit

Note 1: Group 3 interrogation also returns the parameter data (ASDU 110 – threshold, low limit and high limit) for all monitored measured points.

If the highest bit (bit 31) is set, data will be produced by the driver for the specified point at the rate set for periodic data generation. Bit 30 (0x40000000) enables scanning of this point for event generation. If the bit is clear and the data type is set for scanning, events will be generated for the point. If the bit is set, events will not be generated for the point. This feature can be used to select which points will generate events for the controlling station and can get rid of event data that is not important to the application.

9.6 IEC 60870-5-104 Server Interoperability Statement

This companion standard presents sets of parameters and alternatives from which subsets have to be selected to implement particular telecontrol systems. Certain parameter values, such as the number of octets in the COMMON ADDRESS of ASDUs represent mutually exclusive alternatives. This means that only one value of the defined parameters is admitted per system. Other parameters, such as the listed set of different process information in command and in monitor direction allow the specification of the complete set or subsets, as appropriate for given applications. This clause summarizes the parameters of the previous clauses to facilitate a suitable selection for a specific application. If a system is composed of equipment stemming from different manufacturers it is necessary that all partners agree on the selected parameters.

Note: In addition, the full specification of a system may require individual selection of certain parameters for certain parts of the system, such as the individual selection of scaling factors for individually addressable measured values.

The selected parameters should be marked in the white boxes as follows:

☐ Function or ASDU is not used

☒ Function or ASDU is used as standardized (default)

R Function or ASDU is used in reverse mode

B Function or ASDU is used in standard and reverse mode

The possible selection (blank, X, R, B) is specified for each specific clause or parameter.

A black check box indicates that the option cannot be selected in this companion standard.

9.6.1 System or device

☐ System definition

☐ Controlling station definition (Master)

☒ Controlled station definition (Slave)

9.6.2 Application Layer

Transmission mode for application data

Mode 1 (Least significant octet first), as defined in clause 4.10 of IEC 60870-5-4, is used exclusively in this companion standard.

Common Address of ASDU

(System-specific parameter)

☐ One octet ☒ Two octets

Information object address

(System-specific parameter)

☐ One octet ☐ Structured

☐ Two octets ☐ Unstructured

☒ Three octets

Cause of transmission

(System-specific parameter)

☐ One octet ☒ Two octets (with originator address)

Length of APDU

(System-specific parameter, specify the maximum length of the APDU per system)

The Maximum length of the APDU is 246 (default). The maximum length may be reduced by the system.

255 Maximum length of APDU per system

9.6.3 Selection of standard ASDUs

Process information in monitor direction

(Station-specific parameter; mark each Type ID "**X**" if it is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

Type ID	Description	Type
☒ <1>	Single-point information	M_SP_NA_1
☒ <2>	Single-point information with time tag	M_SP_TA_1
☒ <3>	Double-point information	M_DP_NA_1
☒ <4>	Double-point information with time tag	M_DP_TA_1
☒ <5>	Step position information	M_ST_NA_1
☒ <6>	Step position information with time tag	M_ST_TA_1
☒ <7>	Bitstring of 32 bit	M_BO_NA_1
☐ <8>	Bitstring of 32 bit with time tag	M_BO_TA_1
☒ <9>	Measured value, normalized value	M_ME_NA_1
☒ <10>	Measured value, normalized value with time tag	M_ME_TA_1
☒ <11>	Measured value, scaled value	M_ME_NB_1
☒ <12>	Measured value, scaled value with time tag	M_ME_TB_1
☒ <13>	Measured value, short floating point value	M_ME_NC_I
☒ <14>	Measured value, short floating point value with time tag	M_ME_TC_1
☒ <15>	Integrated totals	M_IT_NA_1
☒ <16>	Integrated totals with time tag	M_IT_TA_1
☐ <17>	Event of protection equipment with time tag	M_EP_TA_1

Type ID	Description	Type
☐ <18>	Packed start events of protection equipment with time tag	M_EP_TB_1
☐ <19>	Packed output circuit information of protection equipment with time tag	M_EP_TC_1
☐ <20>	Packed single-point information with status change detection	M_PS_NA_1
☐ <21>	Measured value, normalized value without quality description	M_ME_ND_1
☒ <30>	Single-point information with time tag CP56Time2a	M_SP_TB_1
☒ <31>	Double-point information with time tag CP56Time2a	M_DP_TB_1
☒ <32>	Step position information with time tag CP56Time2a	M_ST_TB_1
☐ <33>	Bitstring of 32 bit with time tag CP56Time2a	M_BO_TB_1
☐ <34>	Measured value, normalized value with time tag CP56Time2a	M_ME_TD_1
☒ <35>	Measured value, scaled value with time tag CP56Time2a	M_ME_TE_1
☐ <36>	Measured value, short floating point value with time tag CP56Time2a	M_ME_TF_1
☒ <37>	Integrated totals with time tag CP56Time2a	M_IT_TB_1
☐ <38>	Event of protection equipment with time tag CP56Time2a	M_EP_TD_1
☐ <39>	Packed start events of protection equipment with time tag CP56time2a	M_EP_TE_1
☐ <40>	Packed output circuit information of protection equipment with time tag CP56Time2a	M_EP_TF_1

Process information in control direction

(Station-specific parameter, mark each Type ID '**X**" if it is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

Type ID	Description	Type
☒ <45>	Single command	C_SC_NA_1
☒ <46>	Double command	C_DC_NA_1
☒ <47>	Regulating step command	C_RC_NA_1
☒ <48>	Set point command, normalized value	C_SE_NA_1
☒ <49>	Set point command, scaled value	C_SE_NB_1
☒ <50>	Set point command, short floating point value	C_SE_NC_1
☐ <51>	Bitstring of 32 bit	C_BO_NA_1
☒ <58>	Single command with time tag CP56Time2a	C_SC_TA_1
☒ <59>	Double command with time tag CP56Time2a	C_DC_TA_1
☒ <60>	Regulating step command with time tag CP56Time2a	C_RC_TA_1
☒ <61>	Set point command, normalized value with time tag CP56Time2a	C_SE_TA_1

Type ID	Description	Type
☒ <62>	Set point command, scaled value with time tag CP56Time2a	C_SE_TB_1
☒ <63>	Set point command, short float value with time tag CP56Time2a	C_SE_TC_1
☐ <64>	Bitstring of 32 bit with time tag CP56Time2a	C_BO_TA_1

Either the ASDUs of the set <45>-<51> or of the set <58>-<64> are used.

System information in monitor direction

(Station-specific parameter, mark "**X**" if used)

Type ID	Description	Type
☒ <70>	End of initialization	M_EI_NA_1

System information in control direction

(Station-specific parameter, mark each Type ID '**X**" if it is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

Type ID	Description	Type
☒ <100>	Interrogation command	C_IC_NA_1
☒ <101>	Counter interrogation command	C_CI_NA_1
☒ <102>	Read command	C_RD_NA_1
☒ <103>	Clock synchronization command	C_CS_NA_1
☒ <104>	Test command	C_TS_NB_1
☒ <105>	Reset process command	C_RP_NC_1
☒ <106>	Delay acquisition command	C_DVD_NA_1

Parameter in control direction

(Station-specific parameter, mark each Type ID '**X**" if it is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

Type ID	Description	Type
☒ <110>	Parameter of measured value, normalized value	P_ME_NA_1
☒ <111>	Parameter of measured value, scaled value	P_ME_NB_1
☒ <112>	Parameter of measured value, short floating point value	P_ME_NC_1
☐ <113>	Parameter activation	P_AC_NA_1

File transfer

(Station-specific parameter, mark each Type ID "**X**" if it is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

Type ID	Description	Type
☐ <120>	File ready	F_FR_NA_1
☐ <121>	Section ready	F_SR_NA_1
☐ <122>	Call directory, select file, call file, call section	F_SC_NA_1
☐ <123>	Last section, last segment	F_LS_NA_1
☐ <124>	Ack file, ack section	F_AF_NA_1
☐ <125>	Segment	F_SG_NA_1
☐ <126>	Directory	F_DR_TA_1

9.6.4 Type identifier and cause of transmission assignments

(Station-specific parameters)

Shaded boxes: option not required

Black boxes: option not permitted in this companion standard

Blank boxes: functions or ASDU not used

Mark Type Identification/Cause of Transmission combinations: '**X**" if only used in standard direction, "**R**" if only used in reverse direction, and "**B**" if used in both directions

Type Identification		Cause of transmission																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<1>	M_SP_NA_1		X	X		X						X			X					
<3>	M_DP_NA_1		X	X		X						X			X					
<5>	M_ST_NA_1		X	X		X						X			X					
<7>	M_BO_NA_1		X	X		X						X			X					
<9>	M_ME_NA_1	X		X		X						X			X					
<11>	M_ME_NB_1	X		X		X						X			X					
<13>	M_ME_NC_1	X		X		X						X			X					
<15>	M_IT_NA_1		X	X		X										X				
<20>	M_PS_NA_1																			
<21>	M_ME_ND_1																			
<30>	M_SP_TB_1			X																
<31>	M_DP_TB_1			X																
<32>	M_ST_TB_1			X																
<33>	M_BO_TB_1			X																
<34>	M_ME_TD_1			X																
<35>	M_ME_TE_1			X																
<36>	M_ME_TF_1			X																

Type Identification		Cause of transmission																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<37>	M_IT_TB_1			X												X				
<38>	M_EP_TD_1																			
<39>	M_EP_TE_1																			
<40>	M_EP_TF_1																			
<45>	C_SC_NA_1						X	X	X	X	X						X	X	X	X
<46>	C_DC_NA_1						X	X	X	X	X						X	X	X	X
<47>	C_RC_NA_1						X	X	X	X	X						X	X	X	X
<48>	C_SE_NA_1						X	X	X	X	X						X	X	X	X
<49>	C_SE_NB_1						X	X	X	X	X						X	X	X	X
<50>	C_SE_NC_1						X	X	X	X	X						X	X	X	X
<51>	C_BO_NA_1						X	X			X						X	X	X	X
<58>	C_SC_TA_1						X	X	X	X	X						X	X	X	X
<59>	C_DC_TA_1						X	X	X	X	X						X	X	X	X
<60>	C_RC_TA_1						X	X	X	X	X						X	X	X	X
<61>	C_SE_TA_1						X	X	X	X	X						X	X	X	X
<62>	C_SE_TB_1						X	X	X	X	X						X	X	X	X
<63>	C_SE_TC_1						X	X	X	X	X						X	X	X	X
<64>	C_BO_TA_1						X	X			X						X	X	X	X
<70>	M_EI_NA_1				X															
<100>	C_IC_NA_1						X	X	X	X	X						X	X	X	X
<101>	C_CI_NA_1						X	X			X						X	X	X	X
<102>	C_RD_NA_1					X											X	X	X	X
<103>	C_CS_NA_1						X	X									X	X	X	X
<105>	C_RP_NA_1						X	X									X	X	X	X
<107>	C_TS_TA_1						X	X									X	X	X	X
<110>	P_ME_NA_1						X	X									X	X	X	X
<111>	P_ME_NB_1						X	X									X	X	X	X
<112>	P_ME_NC_1						X	X									X	X	X	X
<113>	P_AC_NA_1																			
<120>	F_FR_NA_1																			
<121>	F_SR_NA_1																			
<122>	F_SC_NA_1																			
<123>	F_LS_NA_1																			
<124>	F_AF_NA_1																			
<125>	F_SG_NA_1																			
<126>	F_DR_TA_1																			
<127>	F_SC_NB_1																			

9.6.5 Basic Application Functions

Station initialization

(Station-specific parameter, mark "**X**" if function is used)

☒ Remote initialization

Cyclic data transmission

(Station-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒ Cyclic data transmission

Read procedure

(Station-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒ Read procedure

Spontaneous transmission

(Station-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒ Spontaneous transmission

Double transmission of information objects with cause of transmission spontaneous

(Station-specific parameter, mark each information type "**X**" where both a Type ID without time and corresponding Type ID with time are issued in response to a single spontaneous change of a monitored object)

The following type identifications may be transmitted in succession caused by a single status change of an information object. The particular Information Object Addresses for which double transmission is enabled are defined in a project-specific list.

☐ Single-point information M_SP_NA_1, M_SP_TA_1, M_SP_TB_1 and M_PS_NA_1

☐ Double-point information M_DP_NA_1, MDP_TA_1 and M_DP_TB_1

☐ Step position information M_ST_NA_1, M_ST_TA_1 and M_ST_TB_1

☐ Bitstring of 32 bit M_BO_NA_1, M_BO_TA_1 and M_BO_TB_1

☐ Measured value, normalized value M_ME_NA_1, M_ME_TA_1, M_ME_ND_1 and M_ME_TD_1

☐ Measured value, scaled value M_ME_NB_1, M_ME_TB_1 and M_ME_TE_1

☐ Measured value, short floating point number M_ME_NC_1, M_ME_TC_1 and M_ME_TF_1

Station Interrogation

(Station-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒	global				
☒	group 1	☒	group 7	☒	group 13
☒	group 2	☒	group 8	☒	group l4
☒	group 3	☒	group 9	☒	group 15
☒	group 4	☒	group 10	☒	group 16
☒	group 5	☒	group 11		
☒	group 6	☒	group 12		

Clock synchronization

(Station-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒ Clock synchronization

☐ Day of week used

☐ RES1, GEN (time tag substituted/not substituted) used

☐ SU-bit (summertime) used (Optional)

Command transmission

(Object-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒ Direct command transmission

☒ Direct set point command transmission

☒ Select and execute command

☒ Select and execute set point command

☒ C_SE_ACTTERM used [note 2]

☒ No additional definition

☒ Short pulse duration (duration determined by a system parameter in the outstation)

☒ Long pulse duration (duration determined by a system parameter in the outstation)

☒ Persistent output

☐ Supervision of maximum delay in command direction of commands and set point commands

60 seconds Maximum allowable delay of commands and set point commands

Transmission of Integrated totals

(Station- or object-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒ Mode A: Local freeze with spontaneous transmission

☐ Mode B: Local freeze with counter interrogation

☐ Mode C: Freeze and transmit by counter-interrogation commands

☒ Mode D: Freeze by counter-interrogation command, frozen values reported spontaneously

☐ Counter read

☒ Counter freeze without reset

☐ Counter freeze with reset

☐ Counter reset

☐ General request counter

☒ Request counter group 1

☒ Request counter group 2

☒ Request counter group 3

☒ Request counter group 4

Parameter loading

(Object-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒ Threshold value

☐ Smoothing factor

☒ Low limit for transmission of measured value

☒ High limit for transmission of measured value

Parameter activation

(Object-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☐ Act/deact of persistent cyclic or periodic transmission of the addressed object

Test procedure

(Station-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☐ Test procedure

File transfer

(Station-specific parameter, mark "**X**" if function is used)

File transfer in monitor direction

- ☐ Transparent file
- ☐ Transmission of disturbance data of protection equipment
- ☐ Transmission of sequence of events
- ☐ Transmission of sequence of recorded analogue values
- ☐ File transfer in control direction
- ☐ Transparent file

Background scan

(Station-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

- ☐ Background scan

Definition of time outs

Parameter	Default Value	Remarks	Selected Value
t_0	60 seconds	Time-out of connection establishment	60 seconds
t_1	15 seconds	Time-out of send or test APDUs	
t_2	10 seconds	Time-out for acknowledges in case of no data messages ($t_2 < t_1$)	
t_3	20 seconds	Time-out for sending test frames in case of a long idle time	

Maximum range of values for configurable time-outs: 1 to 255 seconds, accuracy 1 second. (t1, t2, and t3 only)

Maximum number of outstanding I format APDUs k and latest acknowledge APDUs (w)

Parameter	Default Value	Remarks	Selected Value
k	12 APDUs	Maximum difference receive sequence number to send state variable (Maximum value is 19)	
w	8 APDUs	Latest acknowledge after receiving *w* I format APDUs	

Maximum range of values *k*: 1 to 32767 (2^{15}-1) APDUs, accuracy 1 APDU

Maximum range of values *w*: 1 to 32767 (2^{15}-1) APDUs, accuracy 1 APDU

(Recommendation: *w* should not exceed two-thirds of *k*).

Port number

Parameter	Value	Remarks
Port number	2404	In all cases

RFC 2200 suite

RFC 2200 is an official Internet Standard which describes the state of standardization of protocols used in the Internet as determined by the Internet Architecture Board (IAB). It offers a broad spectrum of actual standards used in the Internet. The suitable selection of documents from RFC 2200 defined in this standard for given projects has to be chosen by the user of this standard.

☒ Ethernet 802.3

☐ Serial X.21 interface

☐ Other selection from RFC 2200:

9.7 PTQ-104S Database Design Forms

This section contains a set of forms that can be used to design the databases required by the module.

9.7.1 M_SP_NA_1, M_DP_NA_1, M_ST_NA_1 and M_IT_NA_1 Form

Point Number	Database Address	Group Assignment

9.7.2 M_ME_NA_1 and M_ME_NB_1 Form

Point Number	Database Address	Group Assignment	Default Deadband

9.7.3 Form for All C_ (Command) Data Types, Except C_RC_NA_1

Point Number	Database Address	Monitor Point #	Monitor Database Address	Require Select

9.7.4 C_RC_NA_1 Form

Point Number	Database Address	Monitor Point #	Monitor Database Address

9.8 Frequently Asked Questions

9.8.1 How fast do the "Backplane Data Exchange" commands run?

The "Backplane Data Exchange" commands will execute one at a time during the I/O service interval of the PLC. What this means is that if you had a list of 10 commands at the end of every PLC scan one command would execute. This would mean that it would take 10 PLC scans to execute the 10 commands contained within the "Backplane Data Exchange" section of the configuration file.

9.8.2 What is the maximum number of words I can transfer with a "Backplane Data Exchange" command?

For command types 1 & 2 you may move up to 130 words with each command. Function 3 is somewhat different in that it provides only 64 words of data movement BUT because it is intended to solve very specialized operations its size must be restricted.

9.8.3 How do I configure the module?

The ProTalk requires a simple text based configuration generated by ProSoft Configuration Builder software to make it operational. For a really quick tutorial on the modules communications with the PLC you should review Module Configuration (page 49). For more in-depth information, refer to Backplane Data Exchange (page 56).

9.8.4 What kind of data transfer rates can I expect between the PLC and the module?

Data transfer rates between the PLC and the module depend on a number of variables, among them the number of words being transferred per command, the amount of other network traffic at the time data is being transferred, and overall processor scan times.

9.8.5 Is a .MDC available for configuration of the Module?

Yes. The included DVD will have a version for both Concept 2.5 and 2.6.

9.8.6 Does the module work in a remote rack?

The module is designed to be located in the chassis with the PLC and will not operate in a remote chassis.

9.8.7 Can I use the module in a hot backup system?

The PTQ-104S firmware version 2.05 and above, now supports the 140CPU67160 Hot Standby processor. Refer to Hot Standby Support (page 101) for setup and configuration instructions.

10 Support, Service & Warranty

In This Chapter

- Contacting Technical Support 191
- Warranty Information 192

10.1 Contacting Technical Support

ProSoft Technology, Inc. (ProSoft) is committed to providing the most efficient and effective support possible. Before calling, please gather the following information to assist in expediting this process:

1. Product Version Number
2. System architecture
3. Network details

If the issue is hardware related, we will also need information regarding:

1. Module configuration and associated ladder files, if any
2. Module operation and any unusual behavior
3. Configuration/Debug status information
4. LED patterns
5. Details about the serial, Ethernet or fieldbus devices interfaced to the module, if any.

Note: *For technical support calls within the United States, an emergency after-hours answering system allows 24-hour/7-days-a-week pager access to one of our qualified Technical and/or Application Support Engineers. Detailed contact information for all our worldwide locations is available on the following page.*

Internet	Web Site: www.prosoft-technology.com/support E-mail address: support@prosoft-technology.com
Asia Pacific (location in Malaysia)	Tel: +603.7724.2080, E-mail: asiapc@prosoft-technology.com Languages spoken include: Chinese, English
Asia Pacific (location in China)	Tel: +86.21.5187.7337 x888, E-mail: asiapc@prosoft-technology.com Languages spoken include: Chinese, English
Europe (location in Toulouse, France)	Tel: +33 (0) 5.34.36.87.20, E-mail: support.EMEA@prosoft-technology.com Languages spoken include: French, English
Europe (location in Dubai, UAE)	Tel: +971-4-214-6911, E-mail: mea@prosoft-technology.com Languages spoken include: English, Hindi
North America (location in California)	Tel: +1.661.716.5100, E-mail: support@prosoft-technology.com Languages spoken include: English, Spanish
Latin America (Oficina Regional)	Tel: +1-281-2989109, E-Mail: latinam@prosoft-technology.com Languages spoken include: Spanish, English
Latin America (location in Puebla, Mexico)	Tel: +52-222-3-99-6565, E-mail: soporte@prosoft-technology.com Languages spoken include: Spanish
Brasil (location in Sao Paulo)	Tel: +55-11-5083-3776, E-mail: brasil@prosoft-technology.com Languages spoken include: Portuguese, English

10.2 Warranty Information

For complete details regarding ProSoft Technology's TERMS & CONDITIONS OF SALE, WARRANTY, SUPPORT, SERVICE AND RETURN MATERIAL AUTHORIZATION INSTRUCTIONS please see the documents on the Product CD/DVD or go to www.prosoft-technology/warranty

Documentation is subject to change without notice

Index

%

%SW60 Hot Standby Command Register • 108
%SW61 Hot Standby Status Register • 109

[

[Backplane Configuration] • 54
[C_BO_NA_1 104] • 90
[C_DC_NA_1 104] • 90
[C_RC_NA_1 104] • 90
[C_SC_NA_1 104] • 89
[C_SE_NA_1 104] • 91
[C_SE_NB_1 104] • 91
[C_SE_NC_1 104] • 92
[IEC-870-5-104 Database] • 82
[IEC-870-5-104 IP Addresses] • 86
[IEC-870-5-104] • 72
[M_BO_NA_1 104] • 87
[M_DP_NA_1 104] • 87
[M_IT_NA_1 104] • 89
[M_ME_NA_1 104] • 88
[M_ME_NB_1 104] • 88
[M_ME_NC_1 104] • 88
[M_SP_NA_1 104] • 86
[M_ST_NA_1 104] • 87
[SNTP CLIENT] • 69

A

Add the PTQ Module to the Project • 24, 36
Application Layer • 175

B

Background scan • 184
Backplane Data Exchange • 56, 190
Basic Application Functions • 181
Block 9971
    Block Format for Write • 68
    Set Module Time • 68
Block 9998 or 9999
    Reboot Module • 68
Block Format for Read • 67, 68
Block Format for Write • 66, 67
Build the Project • 25

C

C_RC_NA • 147
C_RC_NA_1 Form • 189
C_SC_NA and C_DC_NA • 146
C_SE_NA and C_SE_NB • 148
C_SE_NC • 149
Cable Connections • 170
Can I use the module in a hot backup system? • 190
Cause of transmission • 176
Clear Queue on Close • 73
Clock synchronization • 182
Command Control Blocks • 57, 61
Command Delay Timer • 75
Command Points Addressing • 145
Command transmission • 182
Common Address of ASDU • 75, 175
Common ASDU Address • 129
Configuring the Processor with Concept • 31
Configuring the Processor with ProWORX • 45
Configuring the Processor with Unity Pro • 21
Connect the PC to the ProTalk Configuration/Debug Port • 16
Connect Your PC to the Processor • 26
Connecting to the Processor with TCPIP • 28
Control Data Transfer • 144
Controlling the Generation of Events • 161
Create a New Project • 22, 33
Cyclic data transmission • 181
Cyclic Data Transmission • 75
Cyclic Set IV Time • 77

D

Data Communication • 150
Data Type Level • 162
Data Type Mapping and Addressing • 58
Database Register • 71
Deadbands • 157
Default Command Qualifier • 83
Definition of time outs • 184
Diagnostics and Troubleshooting • 9, 95, 101, 111
Displaying the Current Page of Registers Again • 117
Does the module work in a remote rack? • 190
Double transmission of information objects with cause of transmission spontaneous • 181
Download the Project to the Module • 96, 98
Download the Project to the Processor • 28, 40, 41

E

Effect on System Scan Time • 104
Error Offset • 74, 75
Error Status Table • 122, 171
Ethernet Configuration • 18
Ethernet Connection • 170
Ethernet LED Indicators • 122
Ethernet Port Configuration - wattcp.cfg • 170
Event Messages (9958) • 65
Event Priority • 77, 167
Event Scan Delay • 76
Events • 65, 156
Exiting the Program • 121

F

Failure Flag Count • 55
File transfer • 179, 184
Form for All C_ (Command) Data Types, Except C_RC_NA_1 • 188
Frequently Asked Questions • 190

Functional Specifications • 125

G

General (All Points) • 161
General Specifications • 124
Group Codes • 87, 89, 92, 150, 174
Group Communication • 150
Guide to the PTQ-104S User Manual • 9

H

Hardware and Software Requirements • 12
Hardware Specifications • 125
Hot Standby Overview • 102
Hot Standby Support • 190
How do I configure the module? • 190
How fast do the • 190
HSBY Operating Modes • 107

I

Identical Configurations • 102
IEC 60870-5-104 Server Interoperability Statement • 175
IEC-870-5-101 Data Types • 58
Implementing Ladder to Support Special Functions • 62
Information for Concept Version 2.6 Users • 32
Information for ProTalk® Product Users • 3
Information object address • 176
Initialize Output Data • 55, 56
Input Data (3x Register Data) • 61
Install ProSoft Configuration Builder Software • 13
Install the ProTalk Module in the Quantum Rack • 14, 15
Installing MDC Configuration Files • 32
Interval for Freeze • 75
Invalid Bit Monitoring • 78, 163
IP Address • 127
Is a .MDC available for configuration of the Module? • 190
IV Check Delay Time • 78
IV Fail Count • 79

K

k (maximum queue) • 73
Keystrokes • 115

L

LED Indicators • 121
Length of APDU • 176
Limitations • 107
Long Pulse Time • 82

M

M_ME_NA • 160
M_ME_NA and M_ME_NB • 141
M_ME_NA_1 and M_ME_NB_1 Form • 187
M_ME_NB • 160
M_ME_NC • 161
M_ME_NC and M_IT_NA • 142
M_SP_NA • 140
M_SP_NA_1, M_DP_NA_1, M_ST_NA_1 and M_IT_NA_1 Form • 186
M_ST_NA • 141
Main Menu • 116
Mapping the Backplane Extensions • 107
Maximum number of outstanding I format APDUs k and latest acknowledge APDUs (w) • 184
Modicon Quantum Hot Standby with Unity and IEC Logic • 104
Module Address • 127
Module Configuration • 49, 190
Monitor Data Transfer • 135
Monitor Direction and Control Direction Point Definition • 130
Monitor Points Addressing • 140
Moving Back Through 5 Pages of Registers • 118
Moving Forward Through 5 Pages of Registers • 118

N

Navigation • 115
Normal Data Transfer Blocks • 61
NTP Server IP Address • 70

O

Opening the Database Menu • 116
Output Data (4x Register Data) • 61
Override Command Qualifier • 83
Overview • 104, 106

P

Package Contents • 12
Parameter activation • 183
Parameter in control direction • 178
Parameter loading • 183
Parameter Offset • 85
PC and PC Software • 13
Performance Considerations • 104
Pinouts • 3, 170
PLC HSBY Command Register • 107
PLC HSBY Status Register • 109
Point Count • 83
Point Level • 162
Port number • 184
Primary and Standby Controllers • 102
Process information in control direction • 177
Process information in monitor direction • 176
Product Specifications • 9, 124
PTQ-104S Database Design Forms • 186
PTQ-104S Protocol Implementation • 127
PTQ-PDPMV1 Hot Standby Considerations • 107

Q

Quantum Hardware • 12

R

Read Module's Time to Processor (9970) • 67
Read procedure • 181
Read Status (9250) • 64, 171
Redisplaying the Menu • 115
Reference • 9, 123
Required Hardware • 112
Returning to the Main Menu • 118, 120
RFC 2200 suite • 185
RS-232 Configuration/Debug Port • 170

S

Scan Events • 79
Select/Operate Timeout • 75
Selection of standard ASDUs • 176
Sequence Flag • 84, 169
Set Module Parameters • 53
Set Priority Queues • 76
Set up Data Memory in Project • 39
Set Up the Project • 50
Setting Up the Modicon Quantum Hot Standby with Unity System • 106
Setting Up the ProTalk Module • 14
Short Pulse Time • 82
SNTP Support • 168
Spontaneous transmission • 181
Standards • 124
Start Here • 9, 11
STARTDT & STOPDT • 153
Station initialization • 181
Station Interrogation • 182
Support, Service & Warranty • 9, 191
System Components • 103
System information in control direction • 178
System information in monitor direction • 178
System or device • 175

T

t0 Connection Timeout • 73
t1 Timeout Set Value • 73
t2 Timeout Set Value • 73
t3 Timeout Set Value • 73
Test procedure • 183
TESTFR Requests • 154
The Configuration/Debug Menu • 112
Time DB Offset • 74
Time Information • 163
Time Type • 80
Time Zone • 70
To configure module parameters • 53
To Create Optional Comment Entries • 93
To print a configuration file • 53, 94
Transmission mode for application data • 175
Transmission of Integrated totals • 183
Type identifier and cause of transmission assignments • 179

U

Understanding System Scan Time in Modicon Quantum Hot Standby with Unity Systems • 104
Use ACTTERM with Setpoint • 76
Use ACTTERM with step • 76
Use Daylight Savings Time • 71
Use IP List • 72
Use Recent • 80
Using Control (Command) Points • 143
Using Monitor Points • 132
Using ProSoft Configuration Builder • 50
Using the Diagnostic Window in ProSoft Configuration Builder • 112

V

Verify Jumper Settings • 14
Viewing Block Transfer Statistics • 116
Viewing Data in ASCII (Text) Format • 118
Viewing Data in Decimal Format • 9, 118
Viewing Data in Floating-Point Format • 118
Viewing Data in Hexadecimal Format • 118
Viewing Module Configuration • 116
Viewing Register Pages • 117
Viewing SNTP Status • 119
Viewing the Next 100 Registers of Data • 118
Viewing the Previous 100 Registers of Data • 118
Viewing Version Information • 119

W

w (latest ack threshold) • 74
Warm Booting the Module • 120
Warnings • 3
Warranty Information • 192
What is the maximum number of words I can transfer with a • 190
What kind of data transfer rates can I expect between the PLC and the module? • 190

Y

Your Feedback Please • 2